Kinetic Monte Carlo Methods

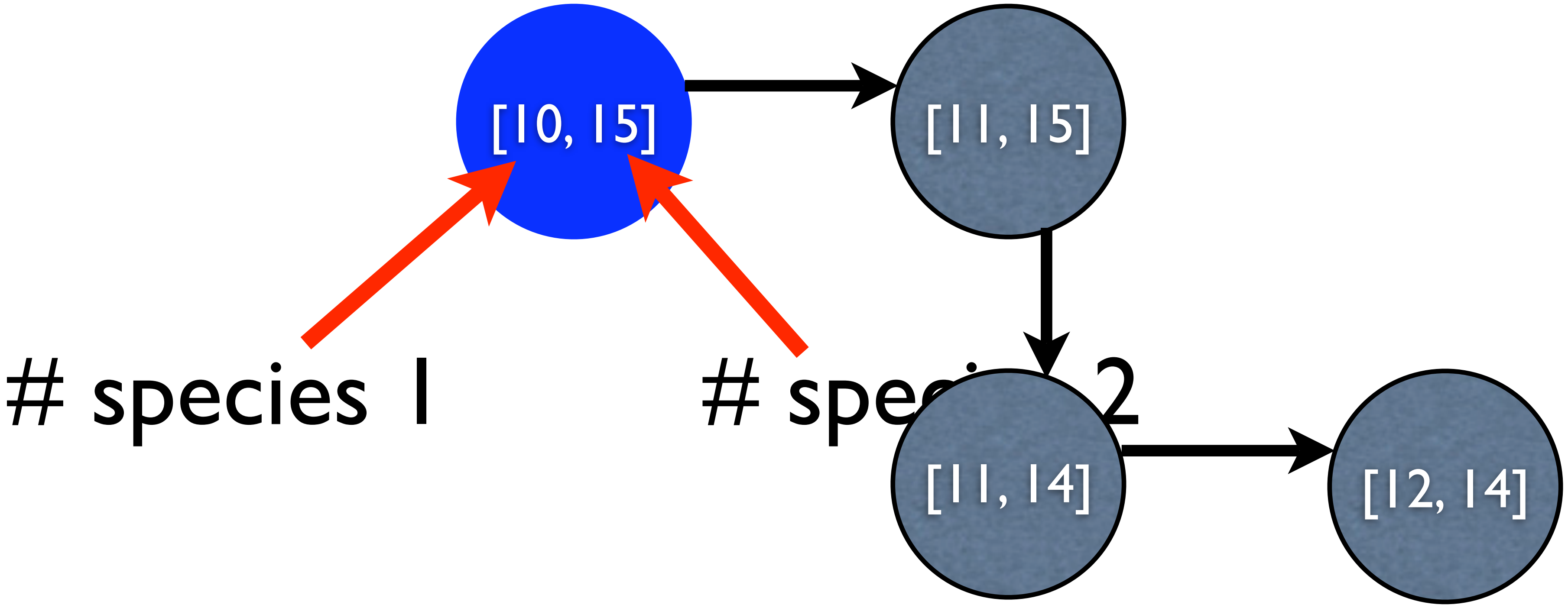
Kinetic Monte-Carlo Simulation Methods

Brian Munsky

- Stochastic Simulation Algorithm
  - D.T. Gillespie, J. Phys. Chem. A **81**, 2340 (1977)
  - M. Gibson and J. Bruck, J. Phys. Chem. **104**, 1876 (2000)
- τ leaping
  - D. Gillespie, J. Chem. Phys. **115**, 1716 (2001); **119**, 8229 (2003)
  - M. Rathinam *et al.*, J. Chem. Phys. **119**, 12784 (2003)
  - T. Tian and K. Burrage, J. Chem. Phys. **121**, 10356 (2004)
  - A. Chatterjee, et al. J. Chem. Phys. **122**, 054104 (2005)
  - Y. Cao, D. Gillespie and L. Petzold, J. Chem. Phys. **123**, 054104 (2005)
- Chemical Langevin Equations
  - D. Gillespie, J. Chem. Phys. **113**, 1716 (2000)
- System Partitioning Methods
  - C. Rao and A. Arkin, J. Chem. Phys. **118**, 4999 (2003)
  - Y. Cao *et al.*, J. Chem. Phys. **122**, 014116 (2005)
- Hybrid Methods
  - E. Haseltine and J. Rawlings, J. Chem. Phys. **117**, 6959 (2002)
  - H. Salis and Y. Kaznessis, J. Chem. Phys. **122**, 054103 (2005)

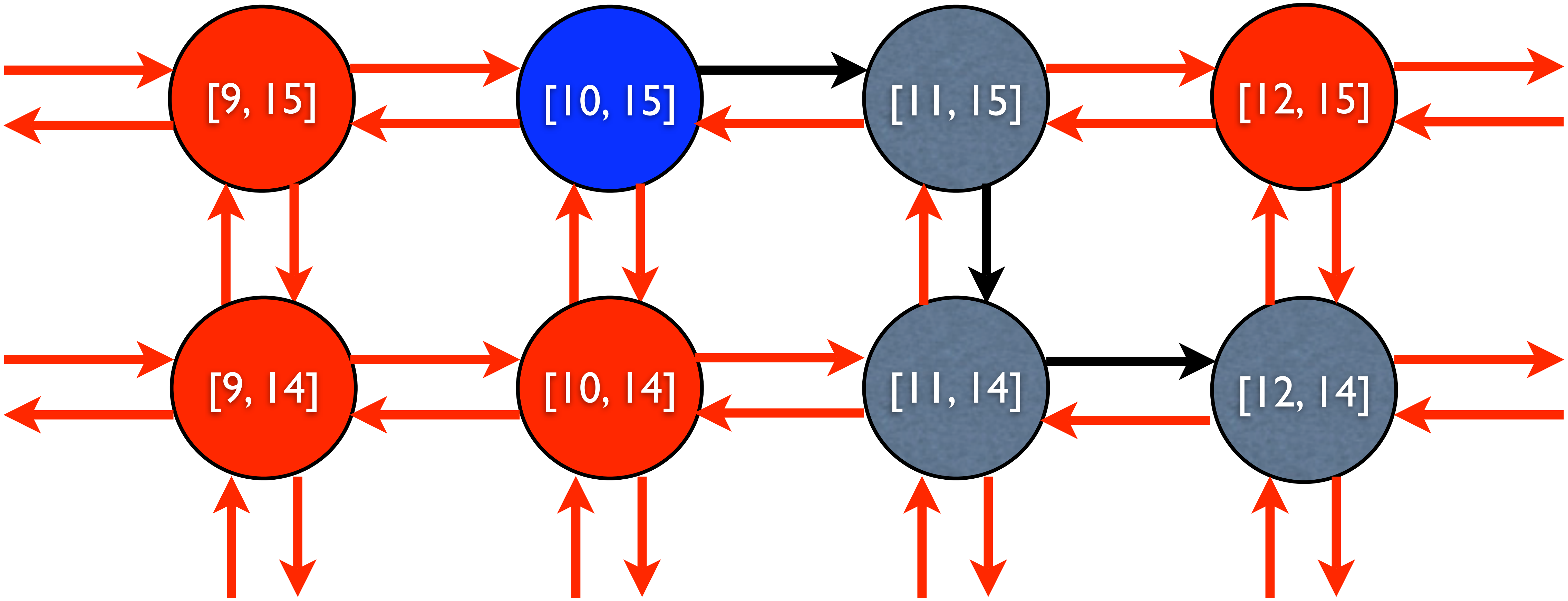
A Jump-Markov description of chemical kinetics

- At any time, the state of the system is defined by its integer population vector: $\mathbf{x} \in \mathbb{Z}^N$
- Reactions are transitions from one state to another:

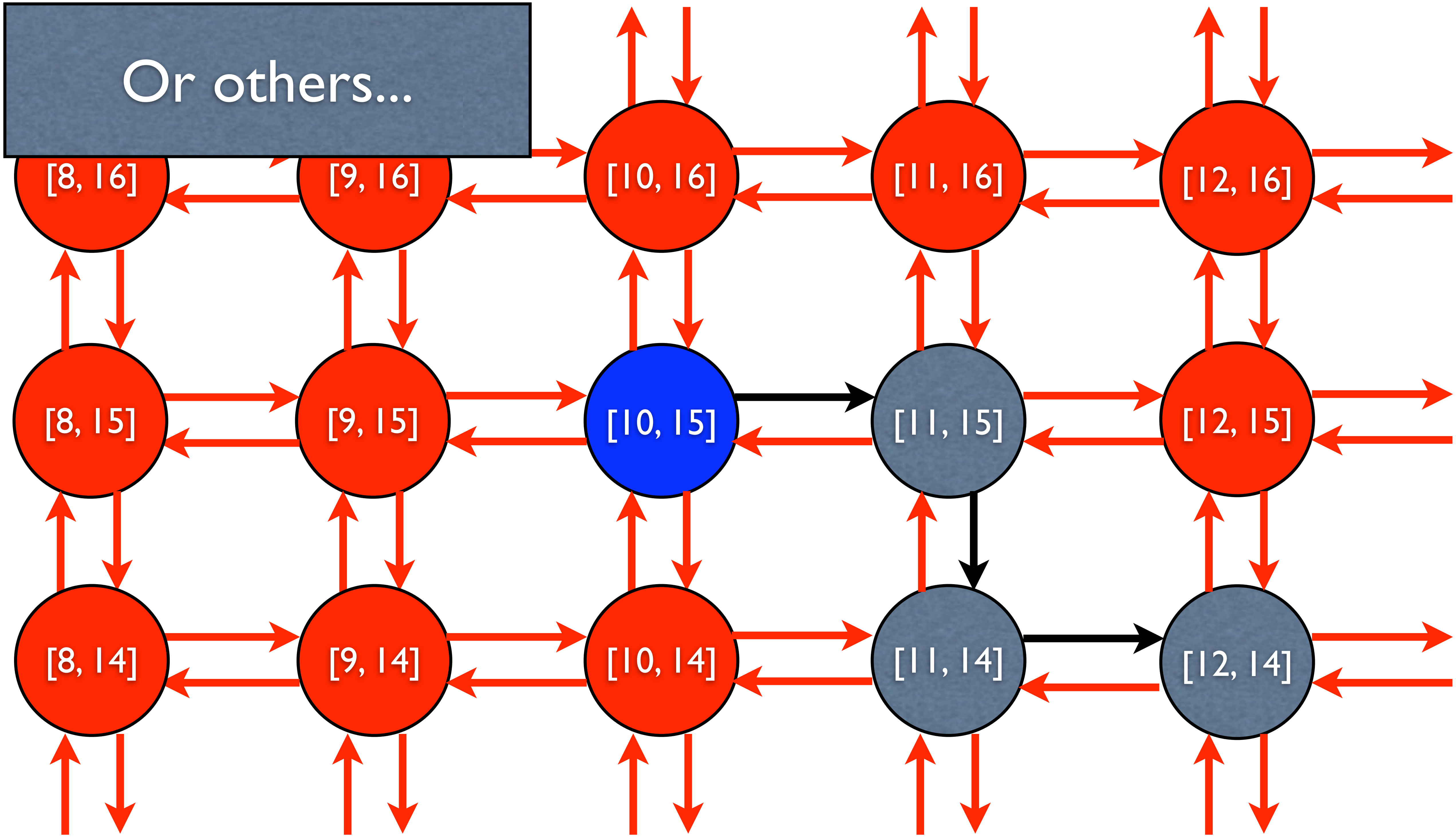


A Jump-Markov description of chemical kinetics

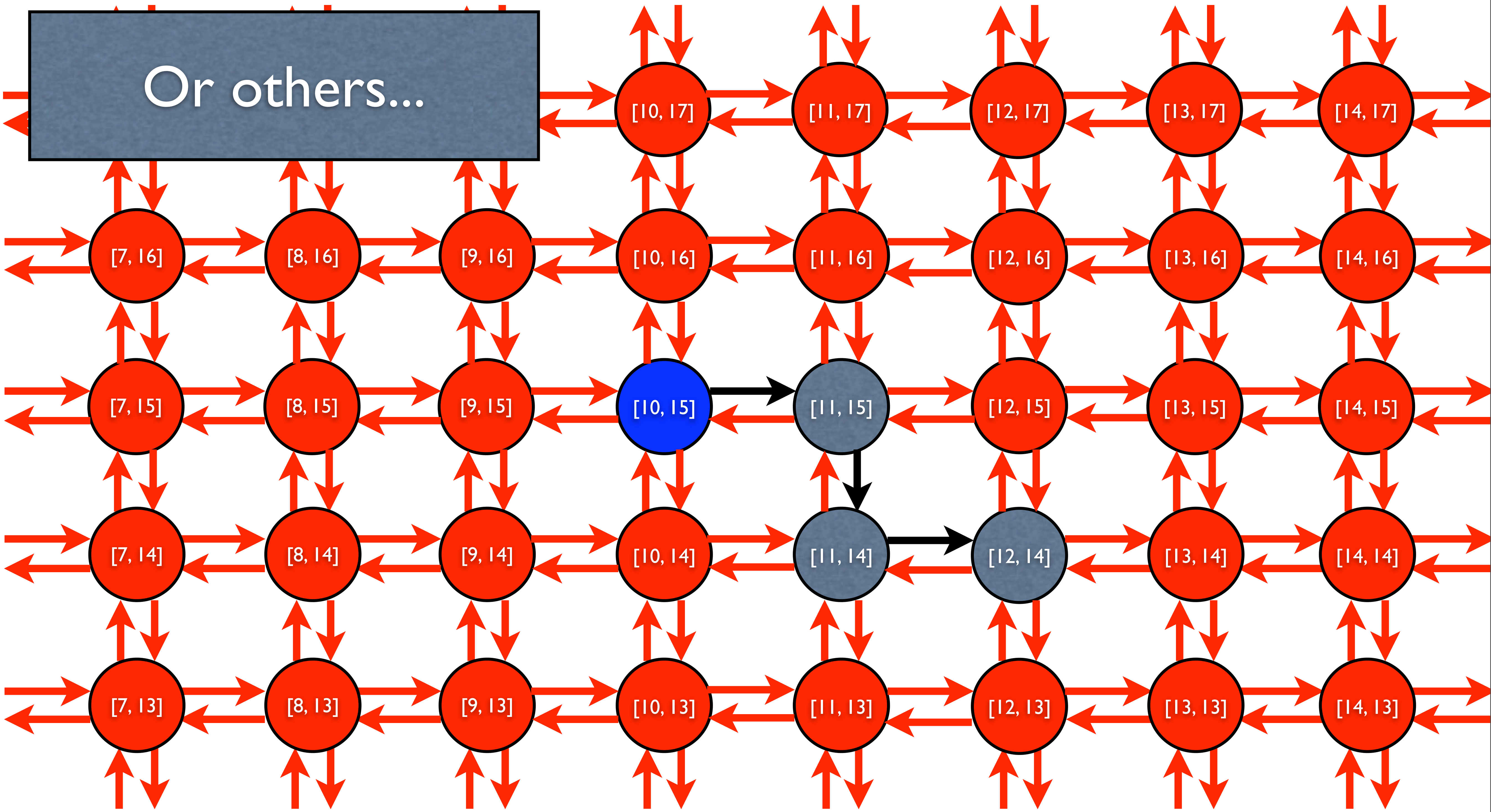
- At any time, the state of the system is defined by its integer population vector: $\mathbf{x} \in \mathbb{Z}^N$
- Reactions are transitions from one state to another:
- These reactions are random, others could have occurred:



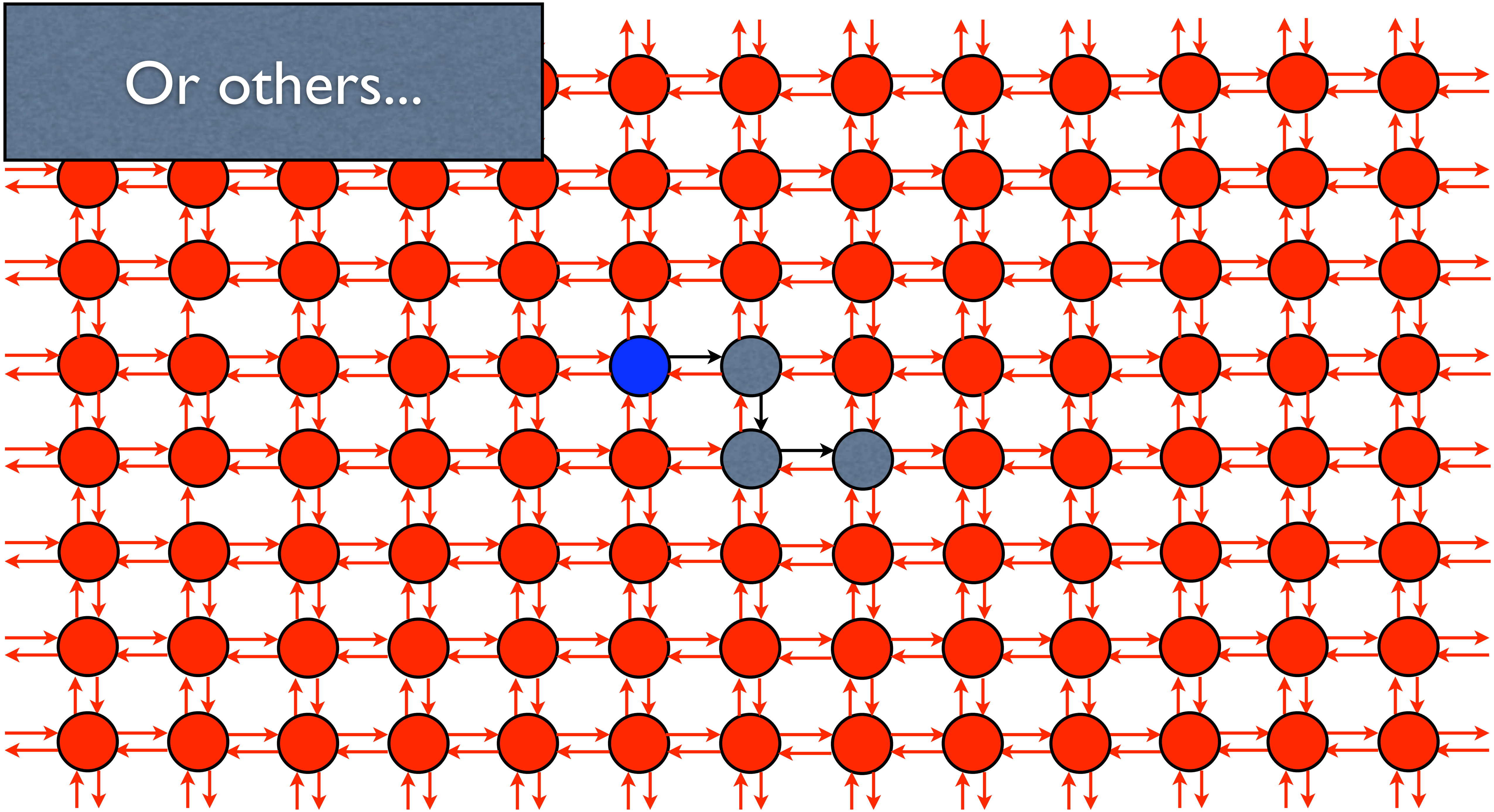
A Jump-Markov description of chemical kinetics



A Jump-Markov description of chemical kinetics



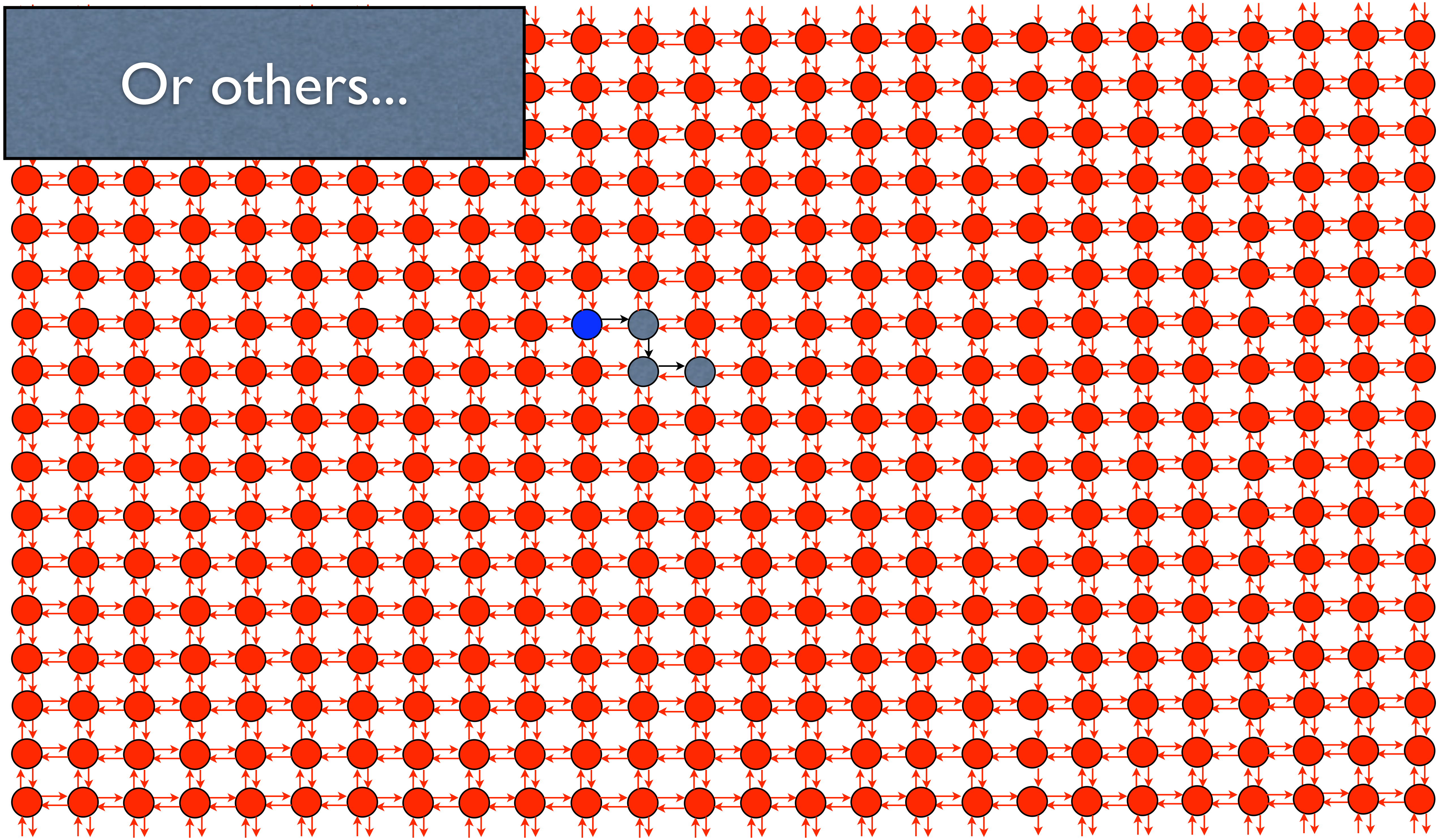
A Jump-Markov description of chemical kinetics



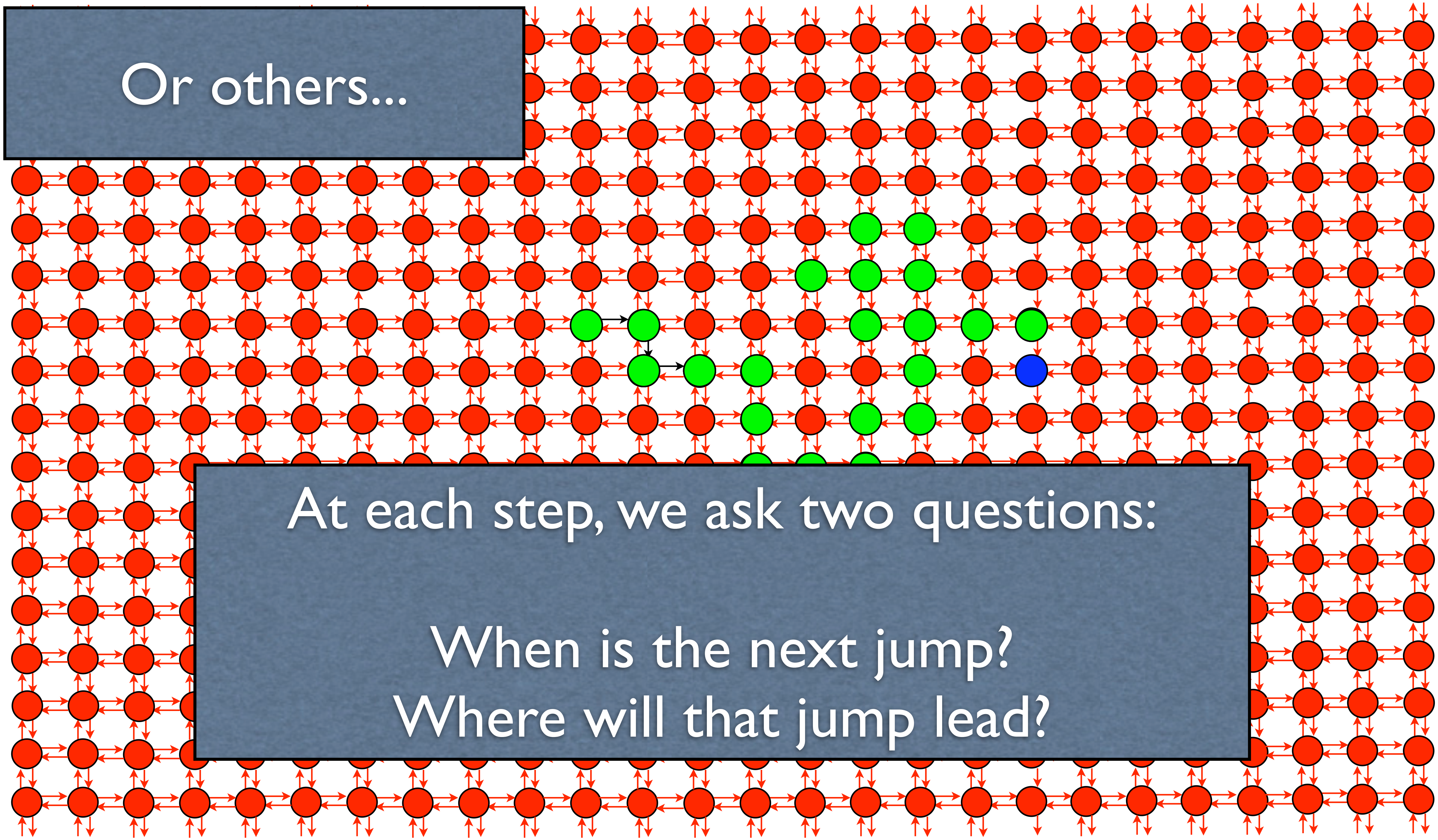
Or others...

A Jump-Markov description of chemical kinetics

Or others...



A Jump-Markov description of chemical kinetics



Or others...

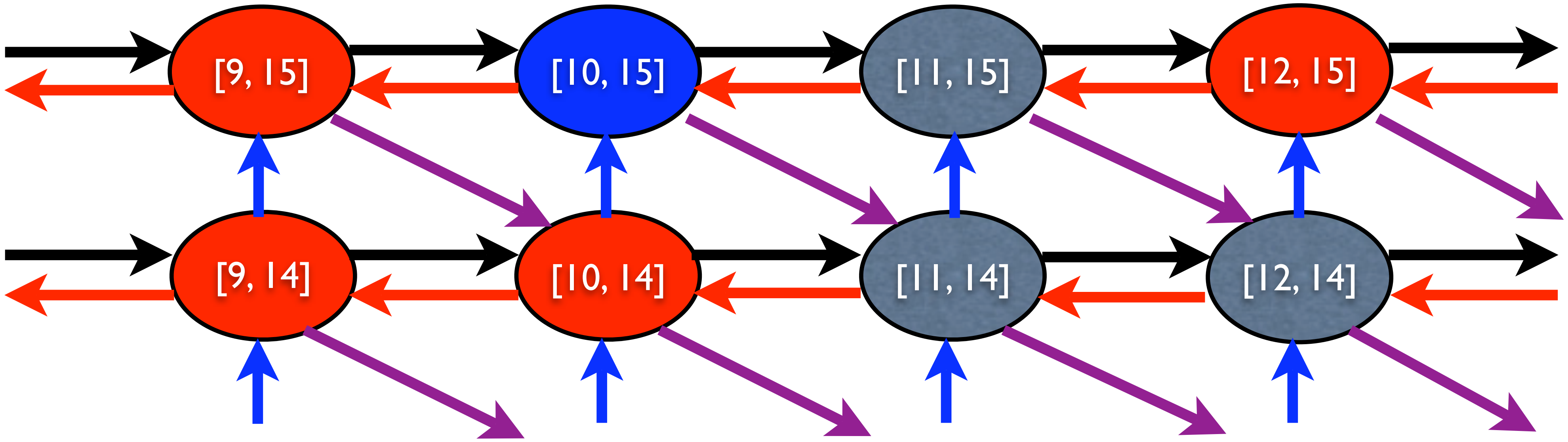
At each step, we ask two questions:

When is the next jump?
Where will that jump lead?

Reaction Stoichiometry (review)

- The Stoichiometric vector, **s**, refers to the relative change in the population vector after a reaction.
- There may be many different reactions for a given stoichiometry.

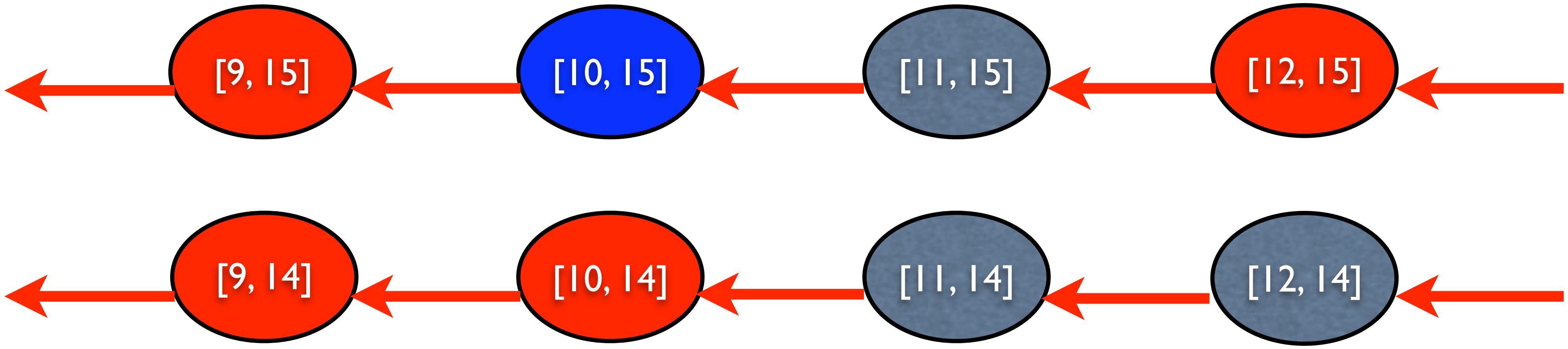
$\mathbf{s}_1 = [1,0]^T$	$\mathbf{s}_2 = [-1,0]^T$	$\mathbf{s}_3 = [0,1]^T$	$\mathbf{s}_4 = [1,-1]^T$
$\mathcal{S}_1 \rightarrow \mathcal{S}_1 + \mathcal{S}_1$	$\mathcal{S}_1 + \mathcal{S}_1 \rightarrow \mathcal{S}_1$	$\mathcal{S}_2 \rightarrow \mathcal{S}_2 + \mathcal{S}_2$	$\mathcal{S}_2 \rightarrow \mathcal{S}_1$
$\mathcal{S}_2 \rightarrow \mathcal{S}_2 + \mathcal{S}_1$	$\mathcal{S}_1 + \mathcal{S}_2 \rightarrow \mathcal{S}_2$	$\mathcal{S}_1 \rightarrow \mathcal{S}_1 + \mathcal{S}_2$	$\mathcal{S}_1 + \mathcal{S}_2 \rightarrow \mathcal{S}_1 + \mathcal{S}_1$
$\emptyset \rightarrow \mathcal{S}_1$	$\mathcal{S}_1 \rightarrow \emptyset$	$\emptyset \rightarrow \mathcal{S}_2$	$\mathcal{S}_2 + \mathcal{S}_2 \rightarrow \mathcal{S}_1 + \mathcal{S}_2$



Reaction Propensities (review)

- The propensity, $\mathbf{w}$, of a reaction is its rate.
- $\mathbf{w}_\mu dt$ is the probability that the μ^{th} reaction will occur in a time step of length dt.
- Typically, propensities depend only upon reactant populations.

$\mathbf{s}_2 = [-1, 0]^T$	$w_2(x_1, x_2)$
$\mathcal{S}_1 + \mathcal{S}_1 \rightarrow \mathcal{S}_1$	$k_1 x_2 (x_1 - 1)/2$
$\mathcal{S}_1 + \mathcal{S}_2 \rightarrow \mathcal{S}_2$	$k_2 x_1 x_2$
$\mathcal{S}_1 \rightarrow \emptyset$	$k_3 x_1$



Exponential Waiting Times

Probability reaction will occur in $[t, t + \Delta t)$: $w\Delta t + \mathcal{O}(\Delta t)^2$

Probability reaction will *not* occur in $[t, t + \Delta t)$: $1 - w\Delta t + \mathcal{O}(\Delta t)^2$

Probability a reaction will not occur in two such time intervals $[t, t + 2\Delta t)$: $\left(1 - w\Delta t + \mathcal{O}(\Delta t)^2\right)^2 = 1 - 2w\Delta t + \mathcal{O}(\Delta t)^2$

Suppose that, $\tau = K\Delta t$, then the probability that no reaction will occur in the interval $[t, t + \tau)$ is

$$\left(1 - w\frac{\tau}{K} + \mathcal{O}(K^{-2})\right)^K$$

Taking the limit as K goes to infinity yields that the probability that no reaction will occur in the interval $[t, t + \tau)$ is

$$\lim_{k \to \infty} \left(1 - w\frac{\tau}{K} + \mathcal{O}(K^{-2})\right)^K = \exp(-w\tau)$$

Exponential Random Variables

The probability that a reaction will occur in the interval $[t, t+\tau)$ is $F_T(\tau) = 1 - \exp(-w\tau)$. This is a cumulative distribution.

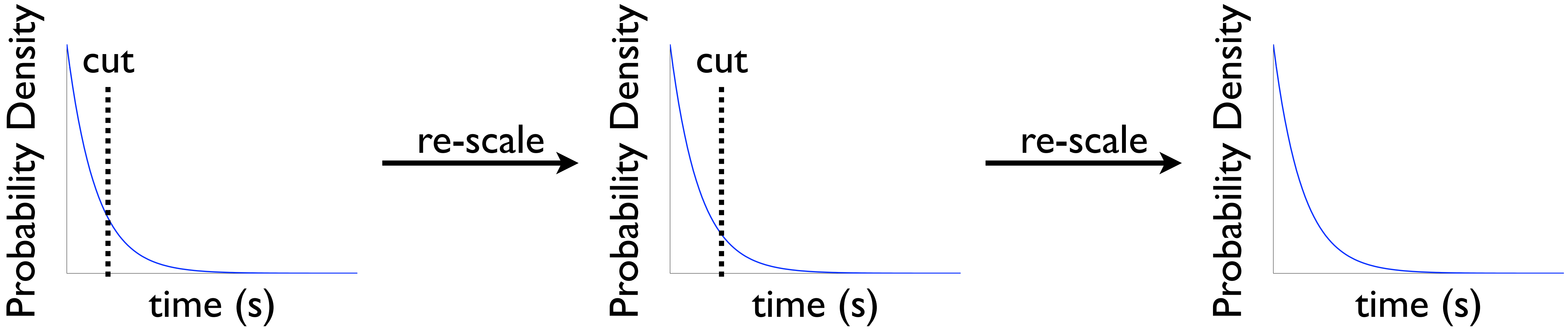
The density (derivative) of the random number, T, is:

$$f_T(\tau) = \frac{1}{w}\exp(-w\tau)$$

Such a random number is known as an *exponentially* distributed random number.

Exponential Waiting Times

- We have assumed that the system is fully described by the population vectors.
- If no reaction occurs, then nothing will have changed.
- Waiting times must be *memoryless* random variables.



- No matter where we cut and scale the distribution, it must always looks the same.

The exponential is the *only* continuous r.v. with this property.

Generating Waiting Times

- To generate an exponentially distributed random number, all we need is a uniform random number generator.

- Find the cumulative distribution,

$$F(t) = 1 - \exp(-\lambda t)$$

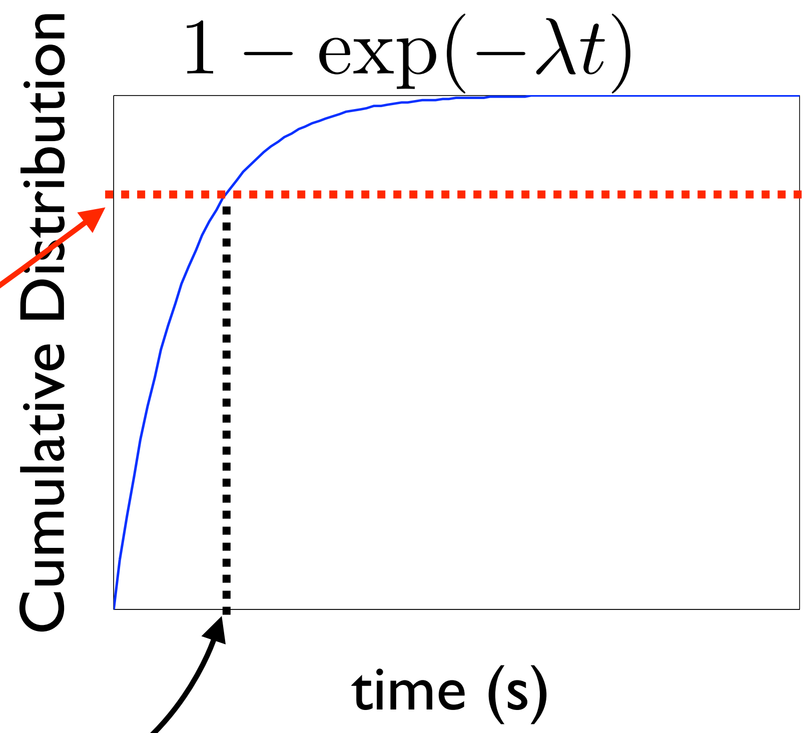
- Generate uniform random number,

$$r \in \mathrm{U}[0, 1]$$

- Find intersection where $F(t) = r$:

$$\tau = \frac{1}{\lambda} \log \frac{1}{1 - r}$$

- This is the time of the next reaction.

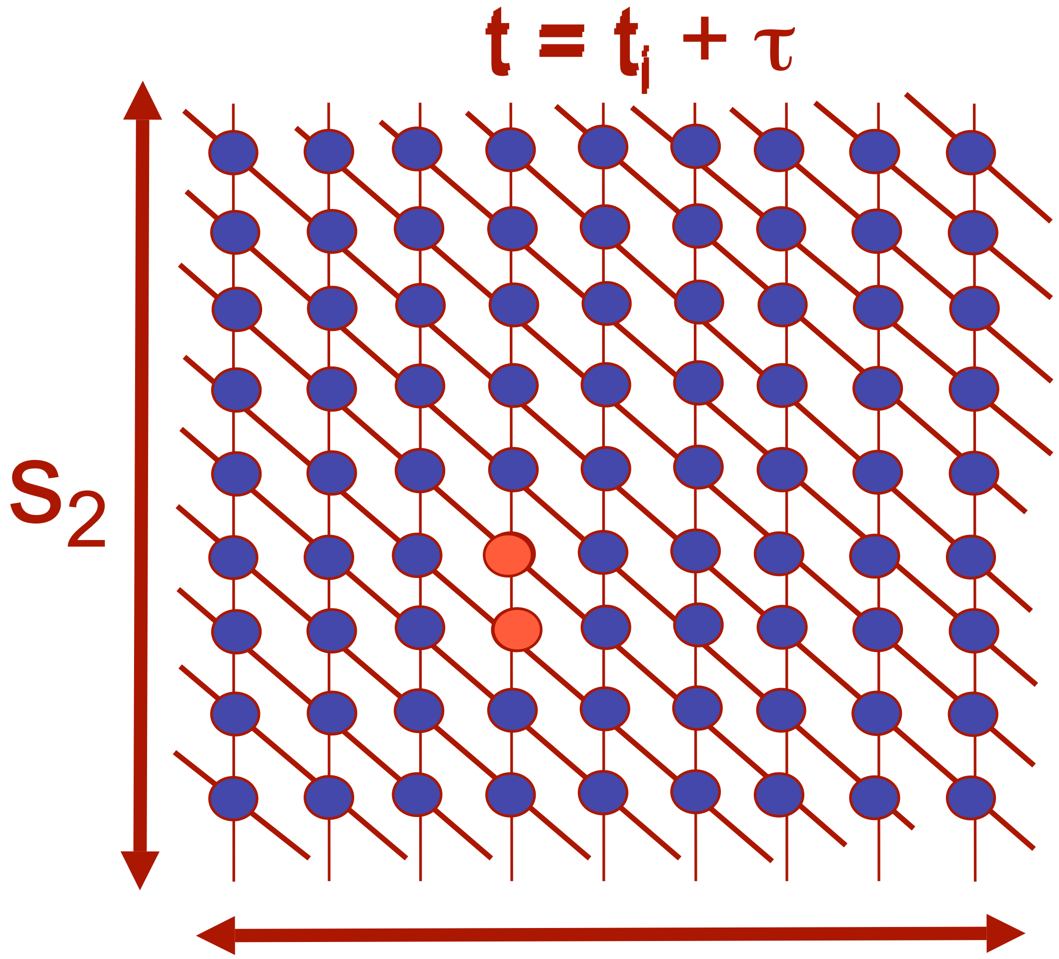


Monte-Carlo Simulation Methods

☑ The Jump Markov Process

- Stochastic Simulation Algorithm
    - D.T. Gillespie, J. Phys. Chem. A **81**, 2340 (1977)
    - M. Gibson and J. Bruck, J. Phys. Chem. **104**, 1876 (2000)

Stochastic Simulation Algorithm



Step 1. Generate the time of the next reaction.

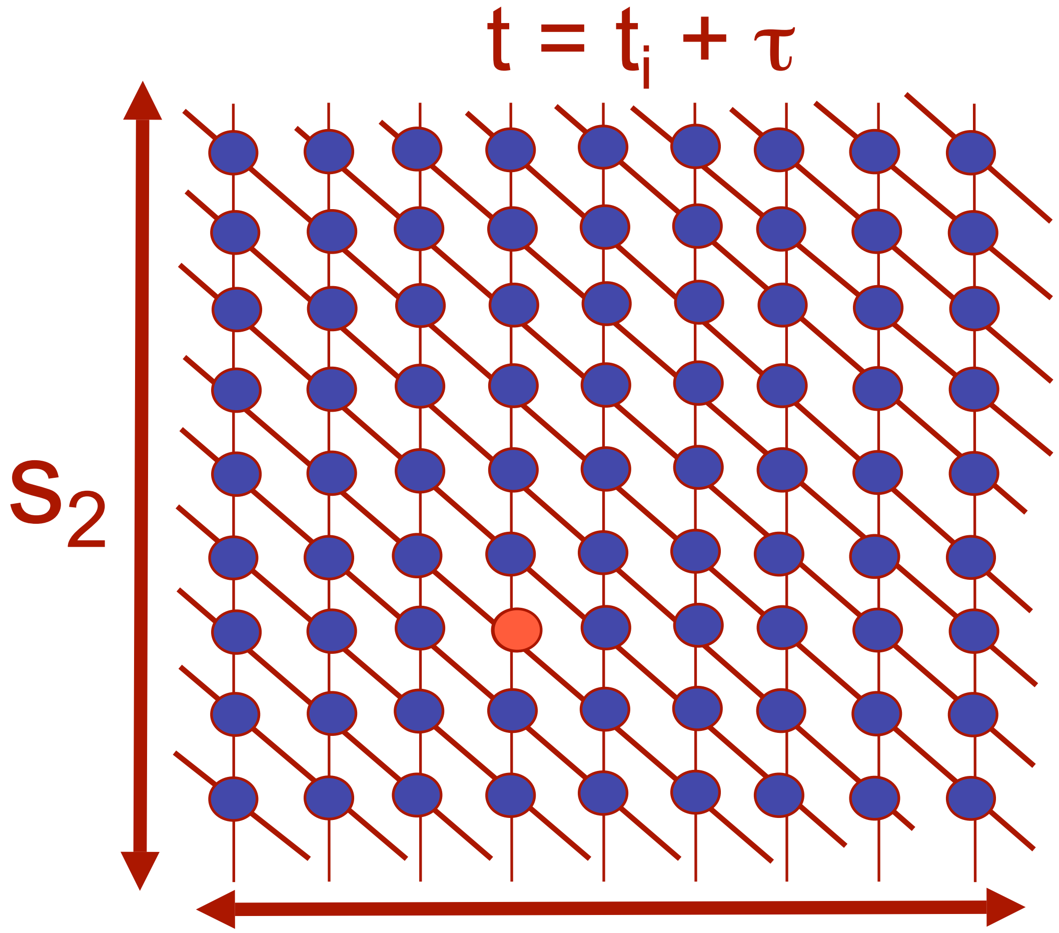
Step 2. Decide which reaction has occurred.

Step 3. Update current Time ($t=t+\tau$) and State ($\mathbf{x} = \mathbf{x}+s_{\mathbf{k}}$).

Monte-Carlo Simulation Methods

- Stochastic Simulation Algorithm
  - D.T. Gillespie, J. Phys. Chem. A **81**, 2340 (1977)
  - M. Gibson and J. Bruck, J. Phys. Chem. **104**, 1876 (2000)
- Possible SSA methods:
  - First Reaction Method (Gillespie ‘77)
    - Next Reaction Method (Gibson and Bruck ‘00)
  - Direct Method (Gillespie ‘77)

The First Reaction Method (FRM)



Step 1. Generate the time of the next reaction of each type.

The time until the next reaction is a random variable of exponential distribution:

$$P_{\tau_\mu}(t) = w_\mu(\mathbf{x})e^{-w_\mu(\mathbf{x})t}$$

To generate each next reaction time, generate r_1 from a uniform distribution on (0,1) and use the equation:

$$\tau_\mu = \frac{1}{w_\mu(\mathbf{x})} \log \frac{1}{r_\mu}$$

Step 2. Decide which reaction has occurred.

This is simply the reaction with the smallest τ_μ:

$$k = \arg\left\{ \min_{\mu \in \{0,\ldots,M\}} \tau_\mu \right\}$$

Step 3. Update current Time (t=t+τ_k) and State ($\mathbf{x} = \mathbf{x} + \mathbf{s}_\mathbf{k}$).

In the FRM each reaction requires M rv’s.

The First Reaction Method SSA in Matlab.

```
clear all
t=0;tstop = 2000;                                  %%specify initial and final times
x = [0; 0];                                        %% Specify initial conditions
S = [1 -1 0  0; 0  0 1 -1];                        %% Specify stoichiometry
w = inline('[10, 1*x(1), 10*x(1), 1*x(2)]','x');   %% Specify Propensity functions
while t<tstop
    tpos = 1./w(x).*log(1./rand(4,1));            % possible times until first reaction
    [tpos,i]=min(tpos);                           % find which is first reaction
    t=t+tpos;
    if t<=t_stop
        x = x+S(:,i);                             % update the configuration
    end
end
```

The Next Reaction Method (NRM)

- In the FRM, we generate times, $\{\tau_\mu\}$, for all *M* reactions and choose the reaction, *k*, with the smallest time, τ_k.
- Only a few species will change population as a result of this reaction--the rest will remain constant.
- For most reactions, the propensity functions will remain constant.
  - For these, the times can be reused in the subsequent step to find the next reaction: $\{\tau_\mu\} \rightarrow \{\ \tau_\mu - \tau_k\}$.
- When there are many different species and reactions, this NRM approach can be done with far fewer random number than the FRM.
- Particularly useful for compartmental or Reaction-Diffusion processes.

Monte-Carlo Simulation Methods

- Stochastic Simulation Algorithm
  - D.T. Gillespie, J. Phys. Chem. A **81**, 2340 (1977)
  - M. Gibson and J. Bruck, J. Phys. Chem. **104**, 1876 (2000)
- Possible SSA methods:
  - First Reaction Method (Gillespie ‘77)
    - Next Reaction Method (Gibson and Bruck ‘00)
  - Direct Method (Gillespie ‘77)

Minimum of two Exponential Random Variables

Let $\{\tau_1, \tau_2, \ldots, \tau_M\}$ be a set of exponentially distributed random variables: $\tau_\mu \in \mathrm{EXP}\left(w_\mu\right)$

The minimum of $\{\tau_\mu\}$ is an exponentially distributed random variable given by:

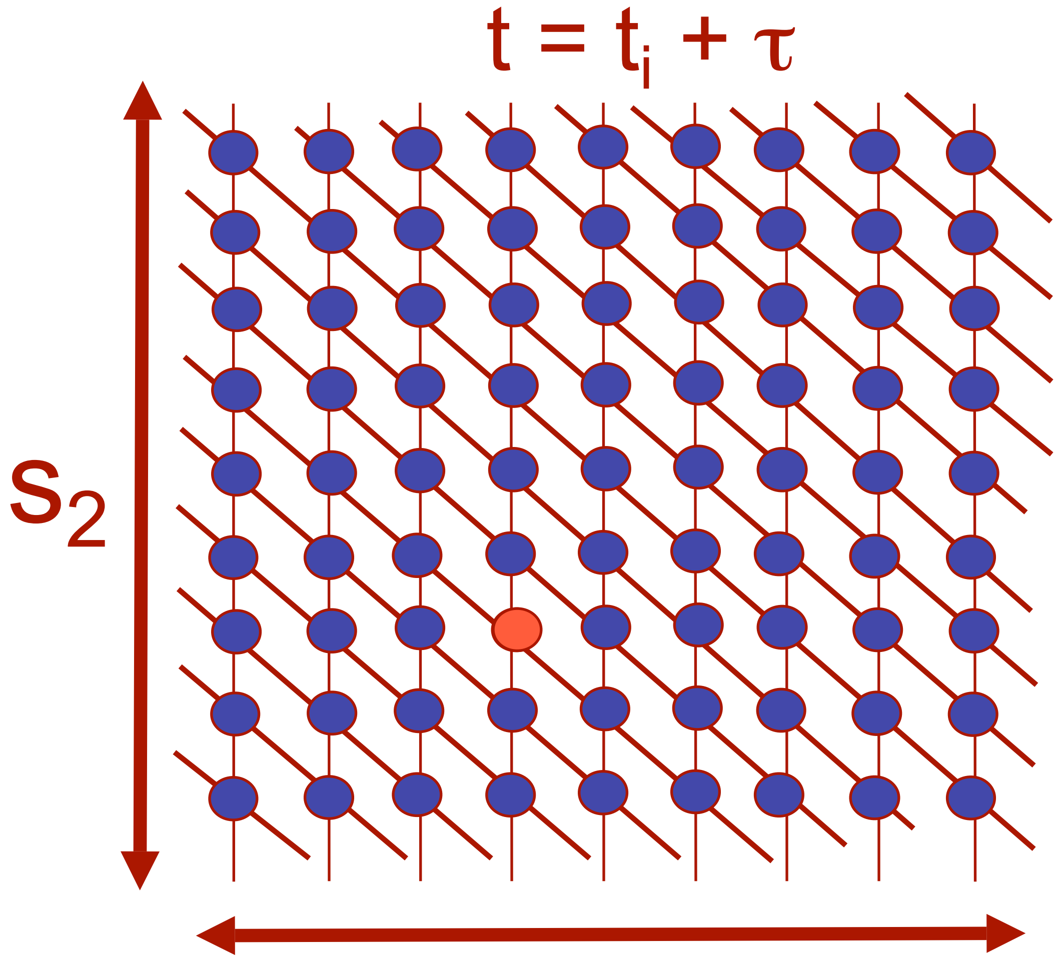
$$\min_{\mu \in \{0,\ldots,M\}} \tau_\mu \in \mathrm{EXP}\left(|\mathbf{w}|_1\right)$$

The argument, *k*, of this distribution is also a random variable with distribution:

$$P(k=\mu) = \frac{w_\mu}{|\mathbf{w}|_1}$$

In the DM we only generate 2 rv's.

The Direct Method (DM)



Step 1. Generate the time of the next reaction.

The time until the next reaction is a random variable of exponential distribution:

$$P_\tau(t) = |\mathbf{w}(\mathbf{x})|_1 \mathrm{e}^{-|\mathbf{w}(\mathbf{x})|_1 t}$$

To generate the next reaction time, generate r_1 from a uniform distribution on (0,1) and use the equation:

$$\tau = \frac{1}{|\mathbf{w}|_1} \log \frac{1}{r_1}$$

Step 2. Decide which reaction has occurred.

To obtain a realization of which reaction will occur, generate a second uniform random number, r_2, and find the smallest *k* such that:

$$\sum_{\mu=1}^{k-1} w_\mu(\mathbf{x}) \leq r_2\,|\mathbf{w}|_1 \leq \sum_{\mu=1}^{k} w_\mu(\mathbf{x})$$

Step 3. Update current Time ($t=t+\tau$) and State ($\mathbf{x} = \mathbf{x}+s_\mathbf{k}$).

The Direct Method SSA in Matlab.

```
clear all
t=0;tstop = 2000;                                  %%specify initial and final times
x = [0; 0];                                        %% Specify initial conditions
S = [1 -1 0  0; 0  0 1 -1];                        %% Specify stoichiometry
w = inline('[10, 1*x(1), 10*x(1), 1*x(2)]','x');   %% Specify Propensity functions
while t<tstop
    w0 = sum(w(x));                               % compute the sum of the prop. functions
    t = t+1/w0*log(1/rand);                       % update time of next reaction
    if t<=t_stop
     r2w0=rand*w0;                  % generate second random number and multiply by prop. sum
     i=1;                                          % initialize reaction counter
     while sum(w(1:i))<r2w0                % increment counter until sum(w(1:i)) exceeds r2w0
       i=i+1;
     end
     x = x+S(:,i);                                 % update the configuration
  end
end
```

Limitations on the SSA

- The SSA is an “exact” simulation of the system.
- But…
  - Stepping through every reaction can take a lot of time.
  - A statistical representation of the system dynamics may require many realizations (10^4 to 10^6).
- Faster approximations are available for some problems.

Monte-Carlo Simulation Methods

- Stochastic Simulation Algorithm (SSA).
- τ-leaping
  - D. Gillespie, J. Chem. Phys. **115**, 1716 (2001)
  - D. Gillespie, L. Petzold, J. Chem. Phys. **119**, 8229 (2003)
  - M. Rathinam *et al.*, J. Chem. Phys. **119**, 12784 (2003)
  - T. Tian and K. Burrage, J. Chem. Phys. **121**, 10356 (2004)
  - Y. Cao, D. Gillespie and L. Petzold, J. Chem. Phys. **123**, 054104 (2005)

τ Leaping

Step 0. Specify length of each time step, τ.

Assume that all propensity functions are constant over the time interval (t,t+τ).

The number of times each reaction will fire is a Poisson* random number with mean $w_\mu\tau$:

$$P_{k_\mu}(n) = \frac{[w_\mu(\mathbf{x})\tau]^n}{n!}\mathrm{e}^{w_\mu(\mathbf{x})\tau}$$

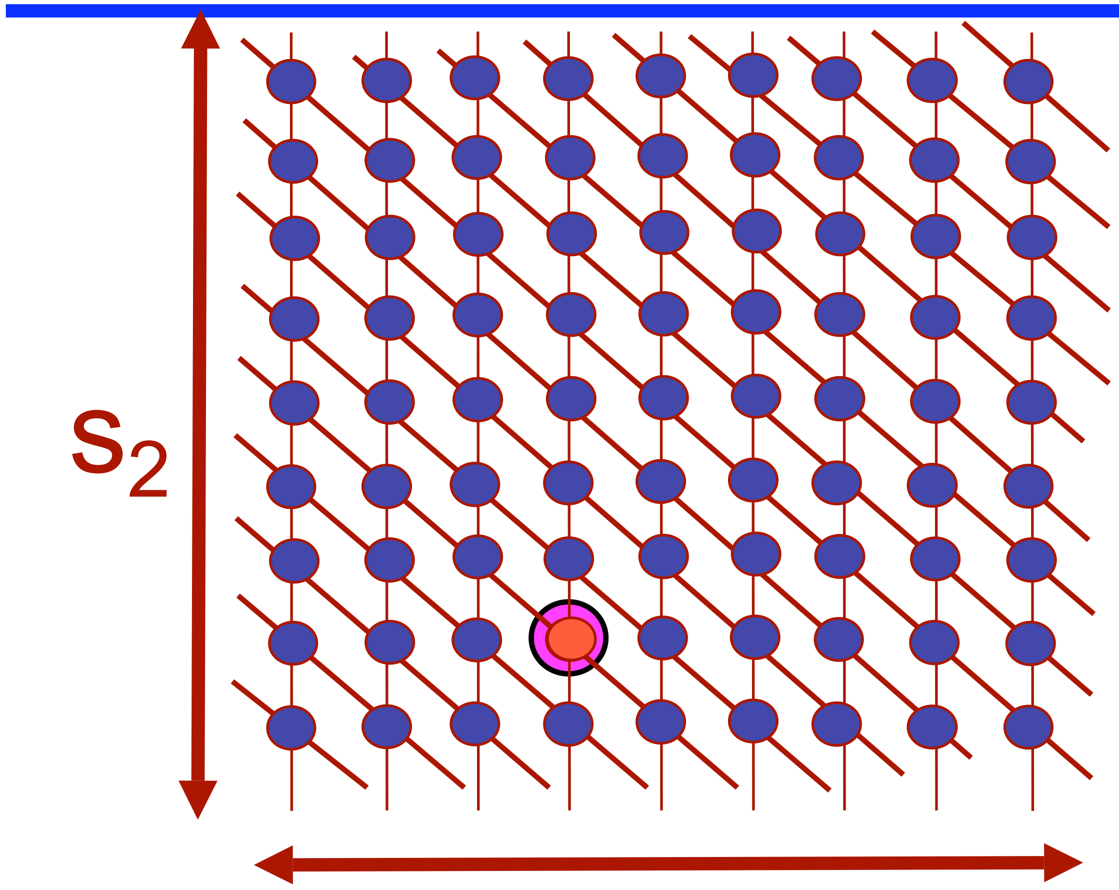
Step 1. For each μ, generate k_μ.

Step 2. Update the time: $t = t + \tau$

Update the state: $\mathbf{x} = \mathbf{x} + \sum_{\mu=1}^{M} k_\mu \mathbf{s}_\mu$

*For some recent studies, binomial RV's are used (T. Tian and K. Burrage, 2004)

τ Leaping



t = t$_i$ +τ Update Time

$$k_1 = 4; \mathbf{s}_1 = [0, 1]^T$$

$$k_2 = 2; \mathbf{s}_1 = [-1, 1]^T$$

$$k_3 = 3; \mathbf{s}_1 = [0, -1]^T$$

$$k_4 = 4; \mathbf{s}_1 = [1, -1]^T$$

The number of times each reaction will fire is a Poisson random number with mean $w_\mu\tau$: $P_{k_\mu}(n) = \frac{[w_\mu(\mathbf{x})\tau]^n}{n!} \mathrm{e}^{w_\mu(\mathbf{x})\tau}$

Step 1. For each μ, generate k_μ.

Step 2. Update the state: $\mathbf{x} = \mathbf{x} + \sum_{\mu=1}^{M} k_\mu \mathbf{s}_\mu$

Update the time: $t = t + \tau$

Limitations of τ leaping

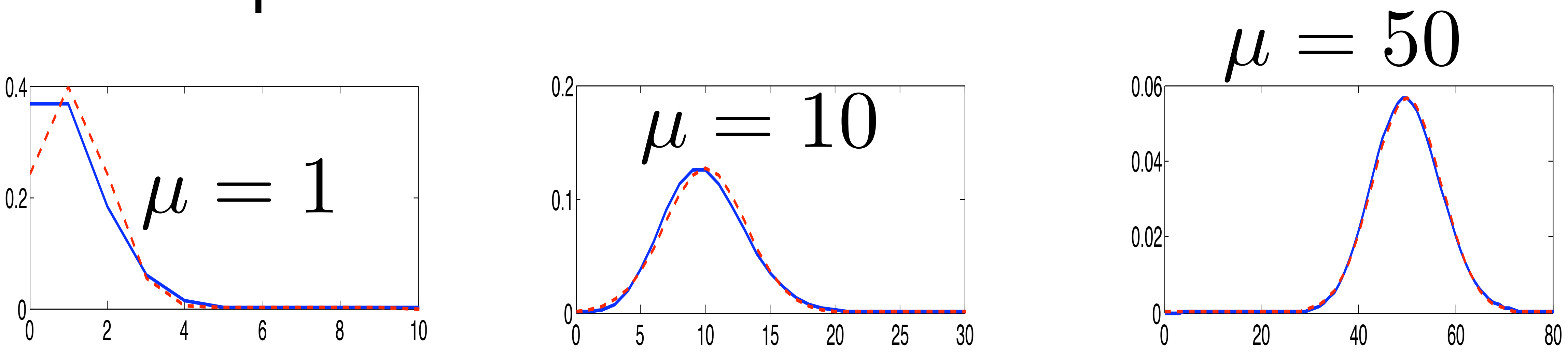
- For many situations τ leaping significantly speeds up the Monte Carlo simulation, but:
  - Poisson r.v.'s are unbounded
  - Propensity functions may change dramatically over small time intervals.
  - May result in negative populations.

Note that these concerns are most important when the population of some species are very small.

Precisely the circumstance where stochastic models are most important!

Chemical Langevin Equation

- Comparison of Poisson and Gaussian random variables.



- For small numbers of reaction steps, tau leaping doesn't give much help.

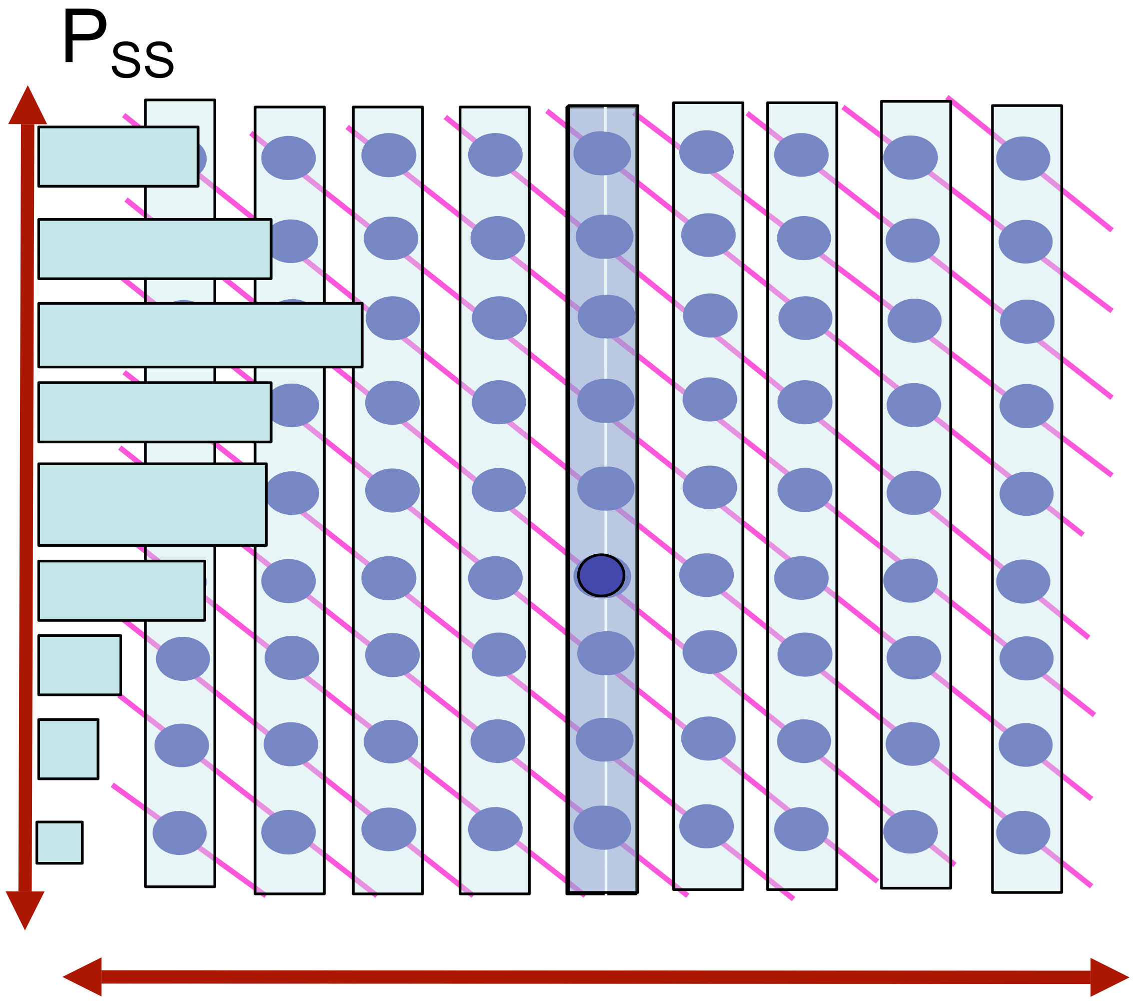
- For large numbers of reactions, replace the Poisson distribution with a normal distribution (same mean and variance. These are cheaper to generate.

- This is known as the chemical Langevin equation.

Monte-Carlo Simulation Methods

- Stochastic Simulation Algorithm (SSA).
- τ-leaping
- System Partitioning Methods
  - Fast--Slow Partitions
    - C. Rao and A. Arkin, J. Chem. Phys. **118**, 4999 (2003)
    - Y. Cao *et al.*, J. Chem. Phys. **122**, 014116 (2005)
  - Continuous--Discrete Partitions
    - E. Haseltine and J. Rawlings, J. Chem. Phys. **117**, 6959 (2002)
    - H. Salis and Y. Kaznessis, J. Chem. Phys. **122**, 054103 (2005)

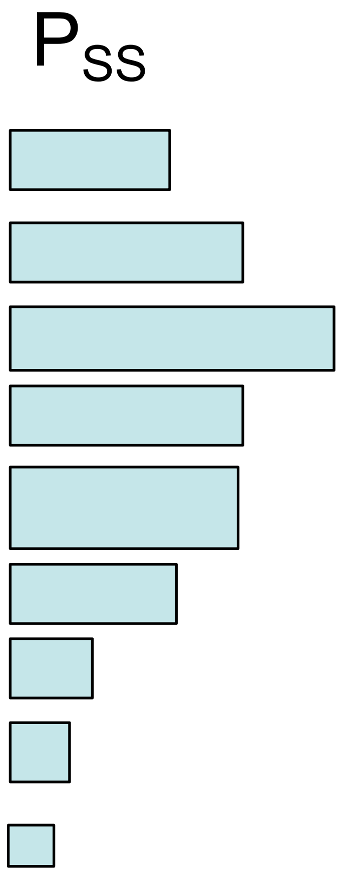
Fast--Slow partitions.



Separate into “fast” and “slow” partitions.

Assume that the “fast” partitions reach probabilistic equilibrium before a slow reaction occurs.

Fast--Slow partitions.



Slow Reaction Propensities

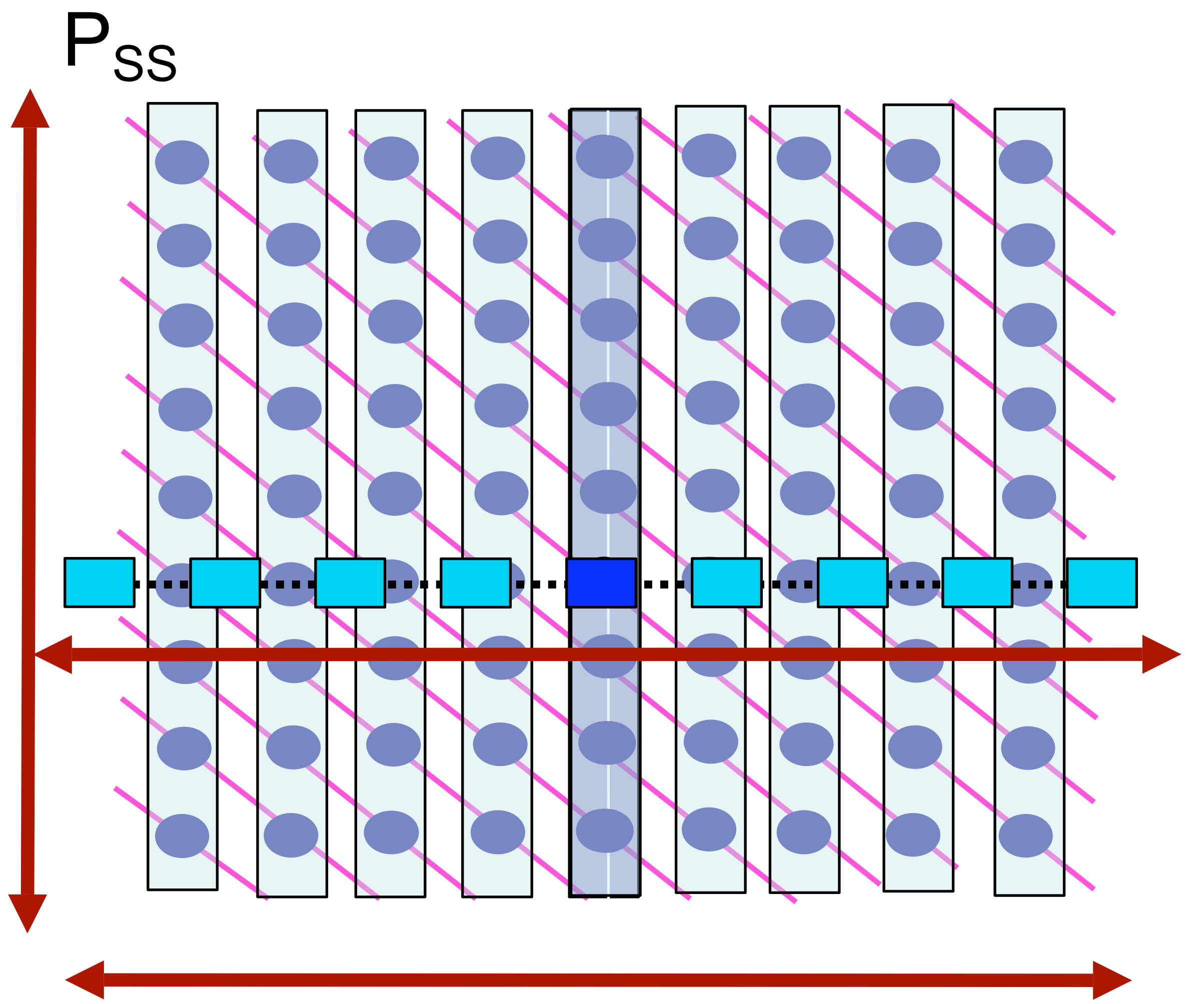
Average Slow Reaction Propensities

$$\text{X} \begin{bmatrix} w_\mu(\mathbf{x}_1) \\ w_\mu(\mathbf{x}_2) \\ w_\mu(\mathbf{x}_3) \\ \vdots \end{bmatrix} = \bar{w}_\mu, \text{ for } \mu = \{1, 2, \ldots, M\}$$

Use the fast sets’ steady state probability distributions to scale the propensity functions of the slow reactions.

Results in a vector of average propensity functions, $\bar{\mathbf{w}}$, for the slow reactions.

Fast--Slow partitions.



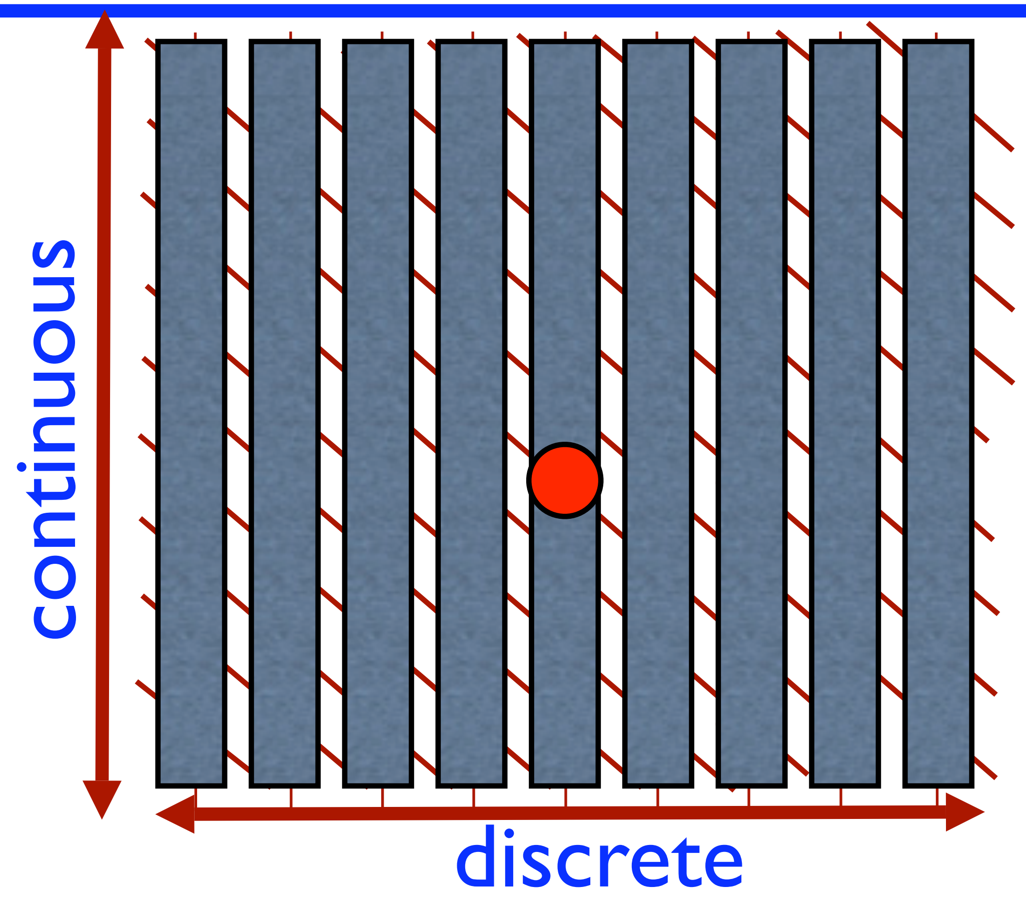
The projection to the slow manifold results in a new lower dimensional Markov chain.

This is simulated with SSA.

Continuous--Discrete partitions.

- In some systems, there are great differences in scale:
  - Large populations (continuous)
  - Small populations (discrete)
- All discrete models take too long.
- All continuous models are inaccurate.
- Hybrid models are necessary.

Separate into “continuous” and “discrete” partitions.

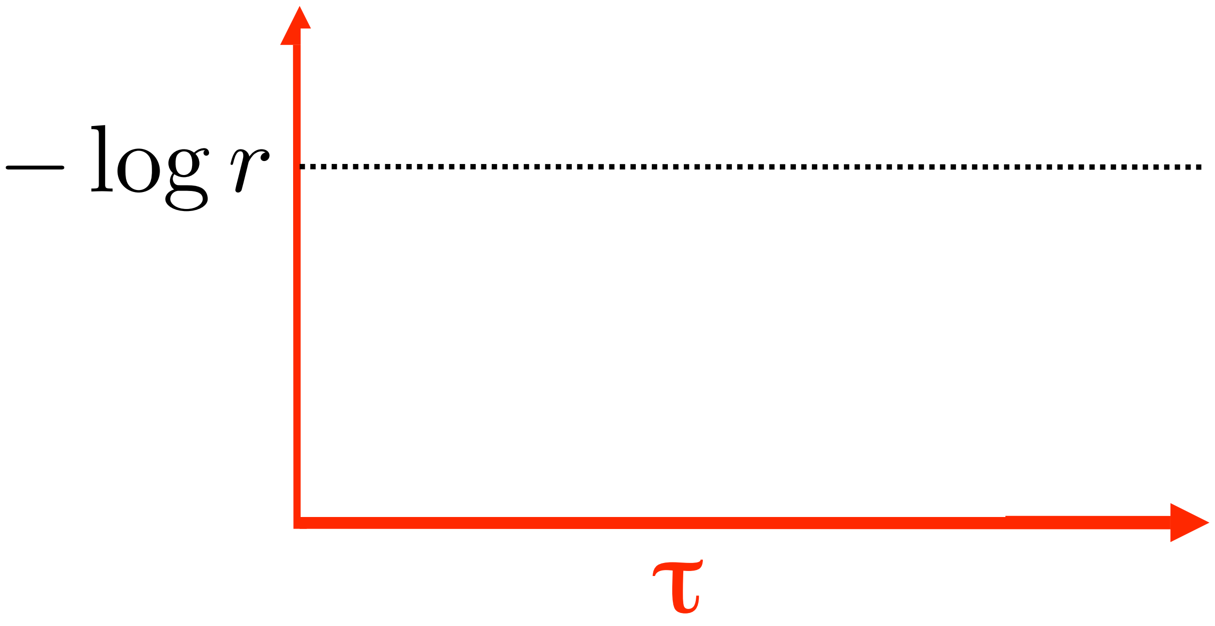


Simulate the continuous part with ordinary or stochastic differential equations.

Choose uniform rv, *r*.

Numerically integrate propensity functions until:

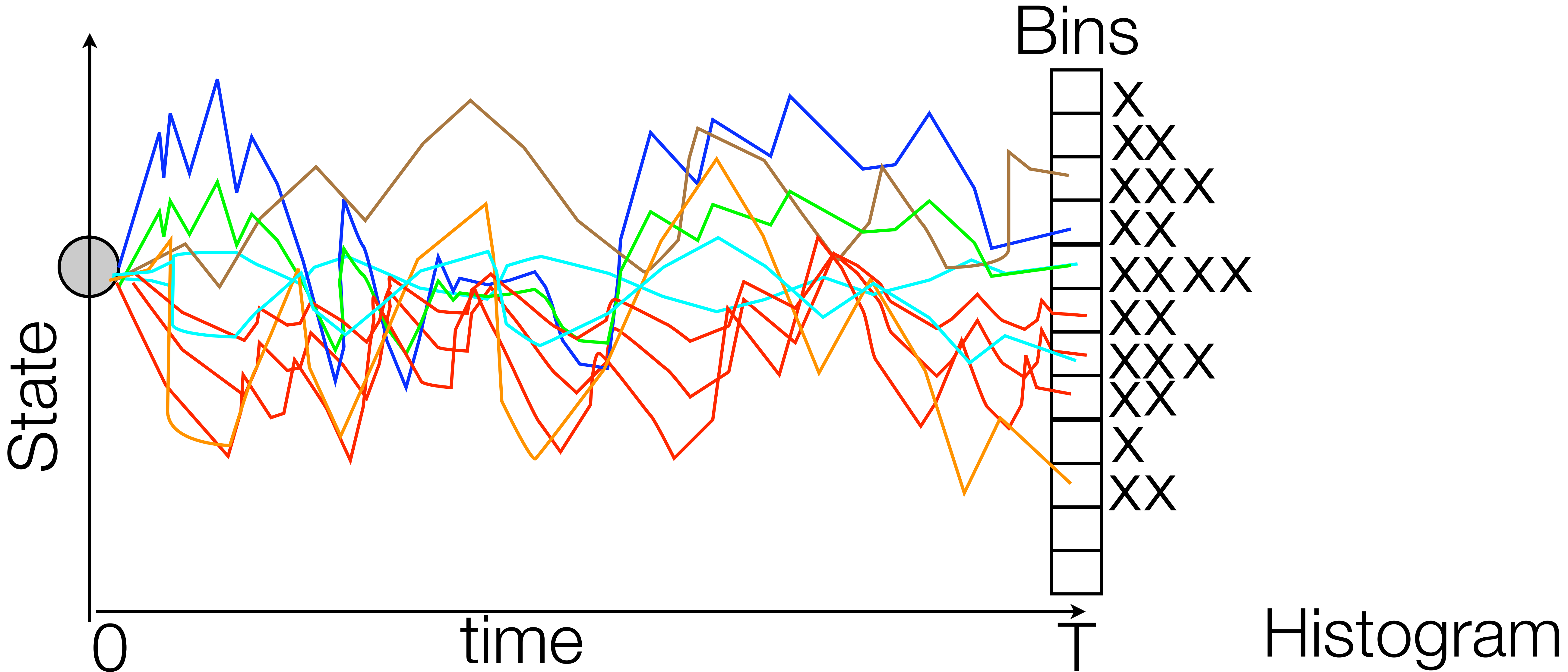
$$\int_{t_0}^{t_0+\tau} \sum_{\mu=1}^{M} w_\mu(\mathbf{x}(t))dt = -\log r$$



Choose next discrete reaction.

Using the SSA to Find Distributions

- The SSA does an excellent job of producing possible trajectories.



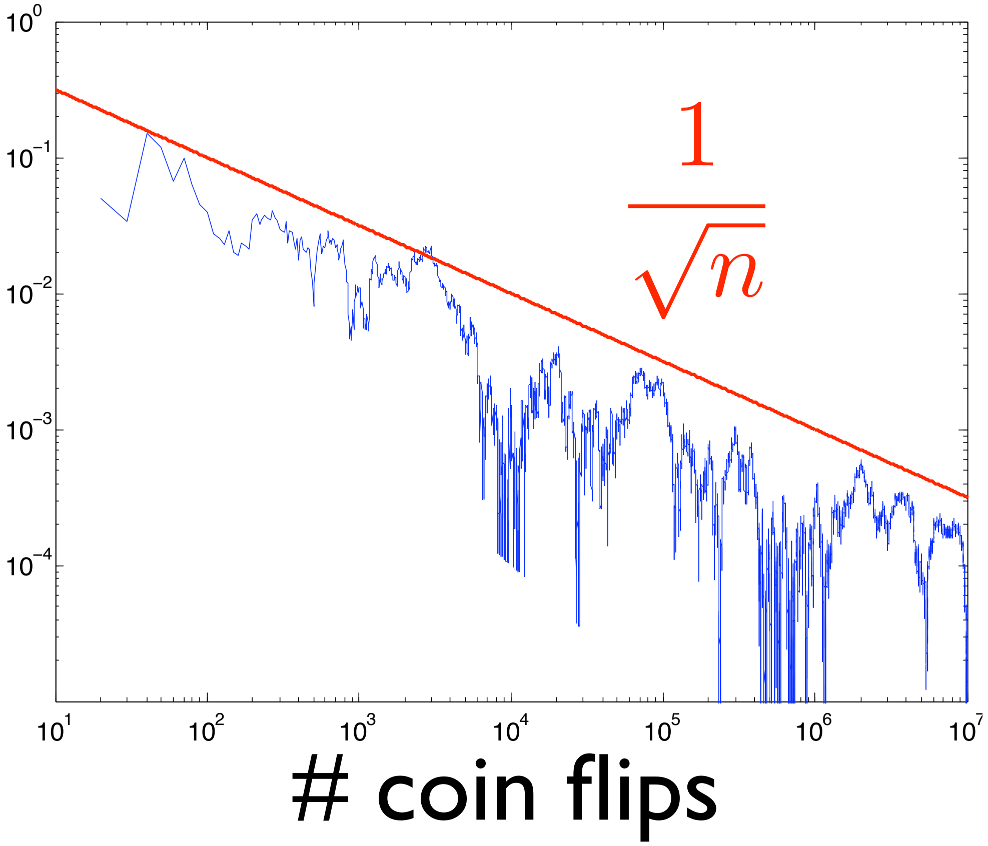
Convergence of the SSA

- To get more accurate distributions, one needs more SSA runs.
- Unfortunately, the convergence rate of any Monte Carlo algorithm is fundamentally limited: $error = \mathcal{O}(n^{-\frac{1}{2}})$
- If very high precision is required, then MC methods will be very inefficient.

Convergence for Coin Toss

error:

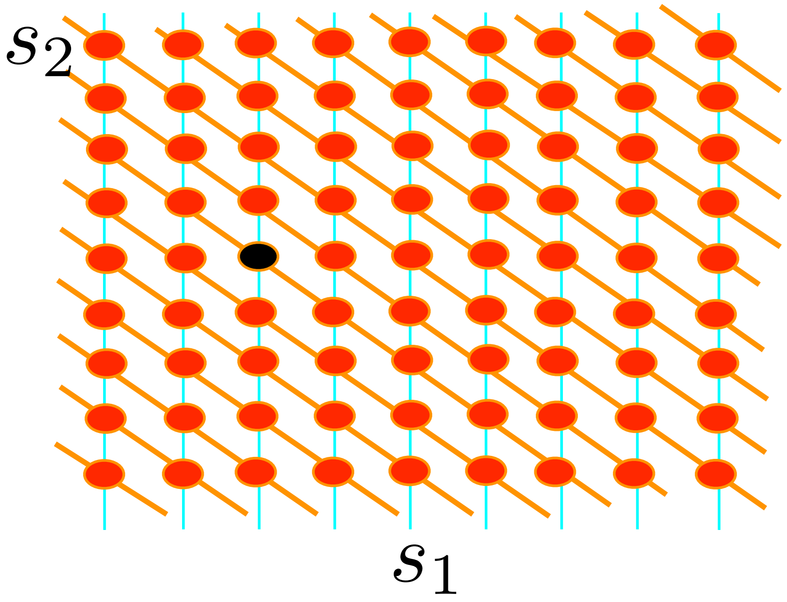
$$\left|\frac{Heads}{n} - 0.5\right|$$



After 10^7 tosses there is still an error of about 3×10^{-4}.

Density Computations using the Finite State Projection

The Chemical Master Equation



The probability that the system is in configuration **x** at *t+dt* is equal to the probability that the system is at **x** at *t*, and no reaction occurs between *t* and *t+dt* plus the probability that the system is one reaction removed from **x** at *t* and that reaction occurs between *t* and *t+dt*.

The CME (McQuarrie '67):

$$\dot{p}(\mathbf{x},t) = -p(\mathbf{x},t)\sum_{k=1}^{M} w_k(\mathbf{x}) + \sum_{k=1}^{M} p(\mathbf{x}-\mathbf{s}_k,t)w_k(\mathbf{x}-\mathbf{s}_k)$$

Define the probability density state vector (pdv): $\mathbf{P}(\mathbf{X},t) := [p(\mathbf{x}_1,t), p(\mathbf{x}_2,t), p(\mathbf{x}_3,t), \ldots]^T$.

$\mathbf{P}(\mathbf{X},t)$ evolves according to the Linear Time Invariant ODE:

$$\dot{\mathbf{P}}(\mathbf{X},t) = \mathbf{A}\cdot\mathbf{P}(\mathbf{X},t)$$

The matrix CME

The Chemical Master Equation

- The solution of the CME is a transfer operator:

$$\mathcal{P}(t_0) \longrightarrow \boxed{\text{CME}} \longrightarrow \mathcal{P}(t_0+\tau)$$

- The dimension of the CME can be INFINITE.
  - Most CME's cannot be solved, so approximations are needed.

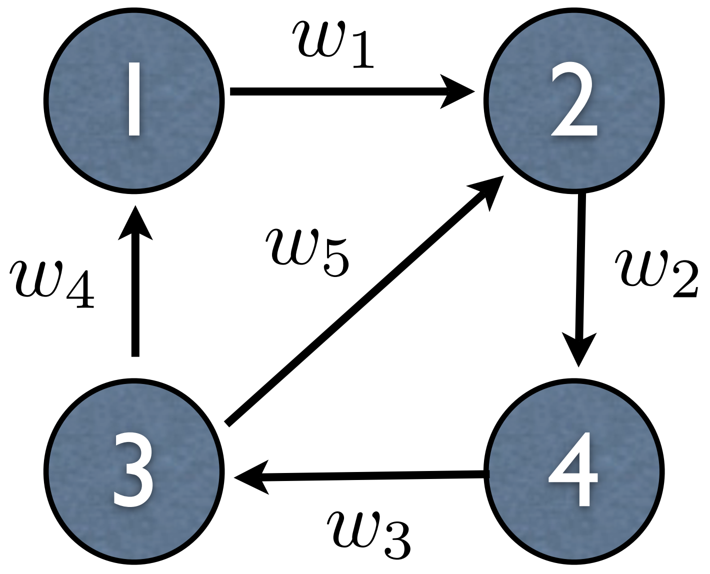
Forming the Generator

A has one row/column for each state.

Each transition, $\mathbf{x}_i \rightarrow \mathbf{x}_j$, contributes to **A** in two locations:

$-w_\mu(\mathbf{x}_i)$ goes in the *diagonal* element $A_{i,i}$

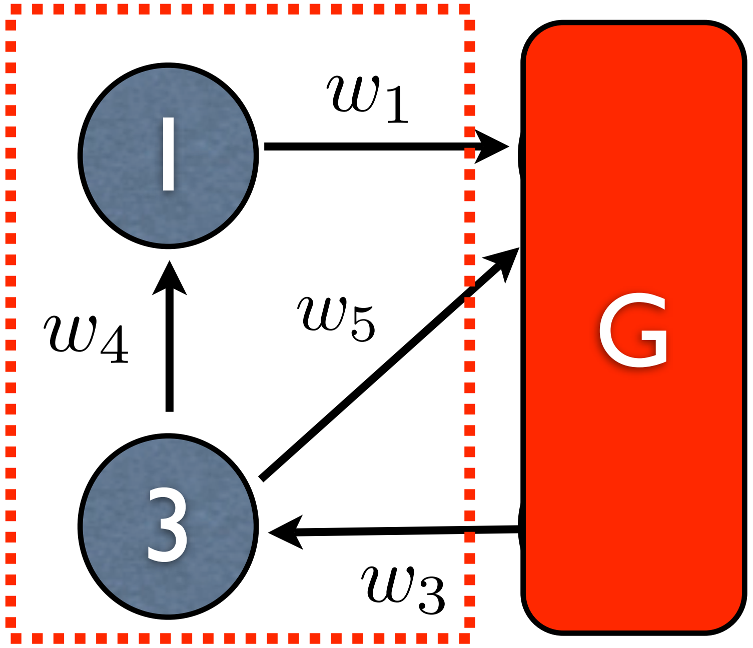
$+w_\mu(\mathbf{x}_i)$ goes in the *off-diagonal* element $A_{j,i}$



$$\mathbf{A} = \begin{bmatrix} \quad \end{bmatrix}$$

The Finite State Projection

Select the states to keep.



Find the corresponding projection matrix:

$$\mathbf{A}_{[1,3]} = \begin{bmatrix} -w_1 & w_4 \\ 0 & -w_4 - w_5 \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} \boxed{-w_1} & 0 & \boxed{w_4} & 0 \\ w_1 & -w_2 & w_5 & 0 \\ \boxed{0} & 0 & \boxed{-w_4 - w_5} & w_3 \\ 0 & w_2 & 0 & -w_3 \end{bmatrix}$$

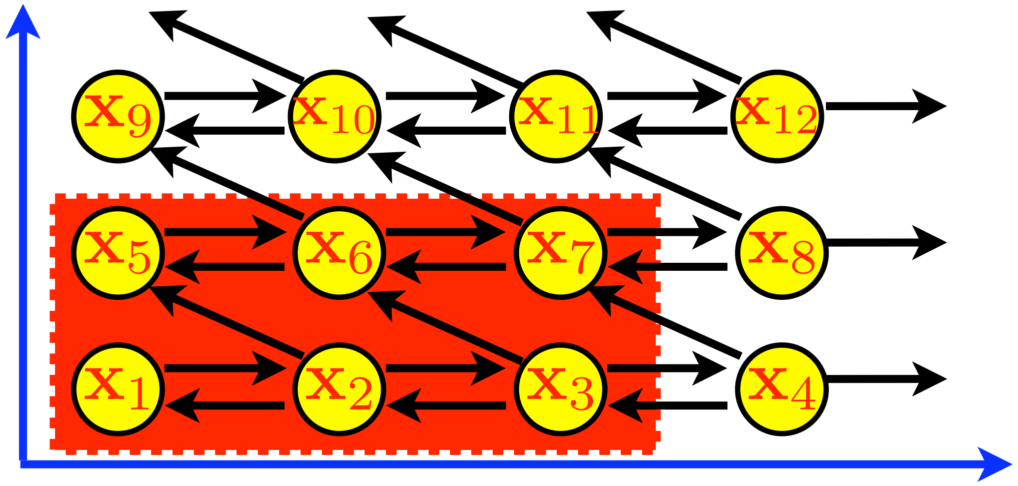
Collapse remaining states into a single absorbing state

$$\mathbf{A}^{FSP}_{[1,3]} = \begin{bmatrix} -w_1 & w_4 & 0 \\ 0 & -w_4 - w_5 & 0 \\ w_1 & w_5 & 0 \end{bmatrix}$$

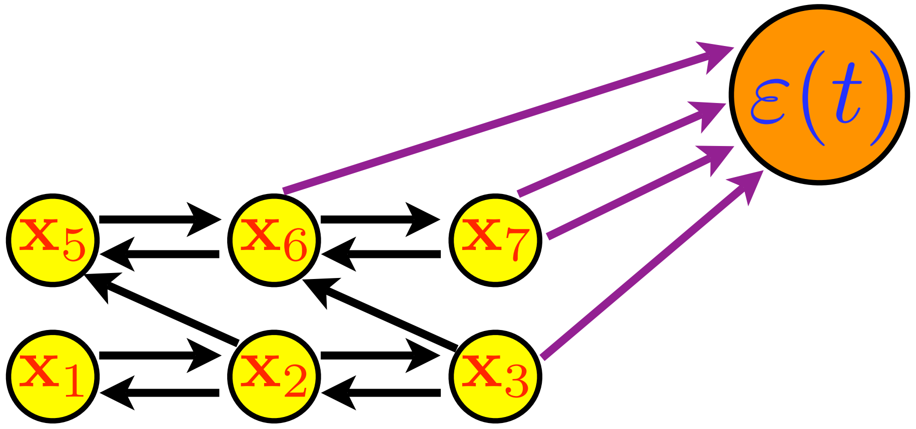
This is the generator for a new Markov chain

The Finite State Projection Method

The Full System



The Projected System (FSP)



Full Master Equation

$$\begin{bmatrix} \dot{\mathbf{P}}_J \\ \dot{\mathbf{P}}_{J'} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_J & \mathbf{A}_{JJ'} \\ \mathbf{A}_{J'J} & \mathbf{A}_{J'} \end{bmatrix} \begin{bmatrix} \mathbf{P}_J(t) \\ \mathbf{P}_{J'}(t) \end{bmatrix}$$

Dimension = $\#(J) + \#(J')$ = Infinite

FSP Master Equation

$$\begin{bmatrix} \dot{\mathbf{P}}_J^{FSP} \\ \dot{\varepsilon} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_J & \mathbf{0} \\ -\mathbf{1}^T\mathbf{A}_J & 0 \end{bmatrix} \begin{bmatrix} \mathbf{P}_J^{FSP}(t) \\ \varepsilon(t) \end{bmatrix}$$

Dimension =$\#(J) + 1$= 7

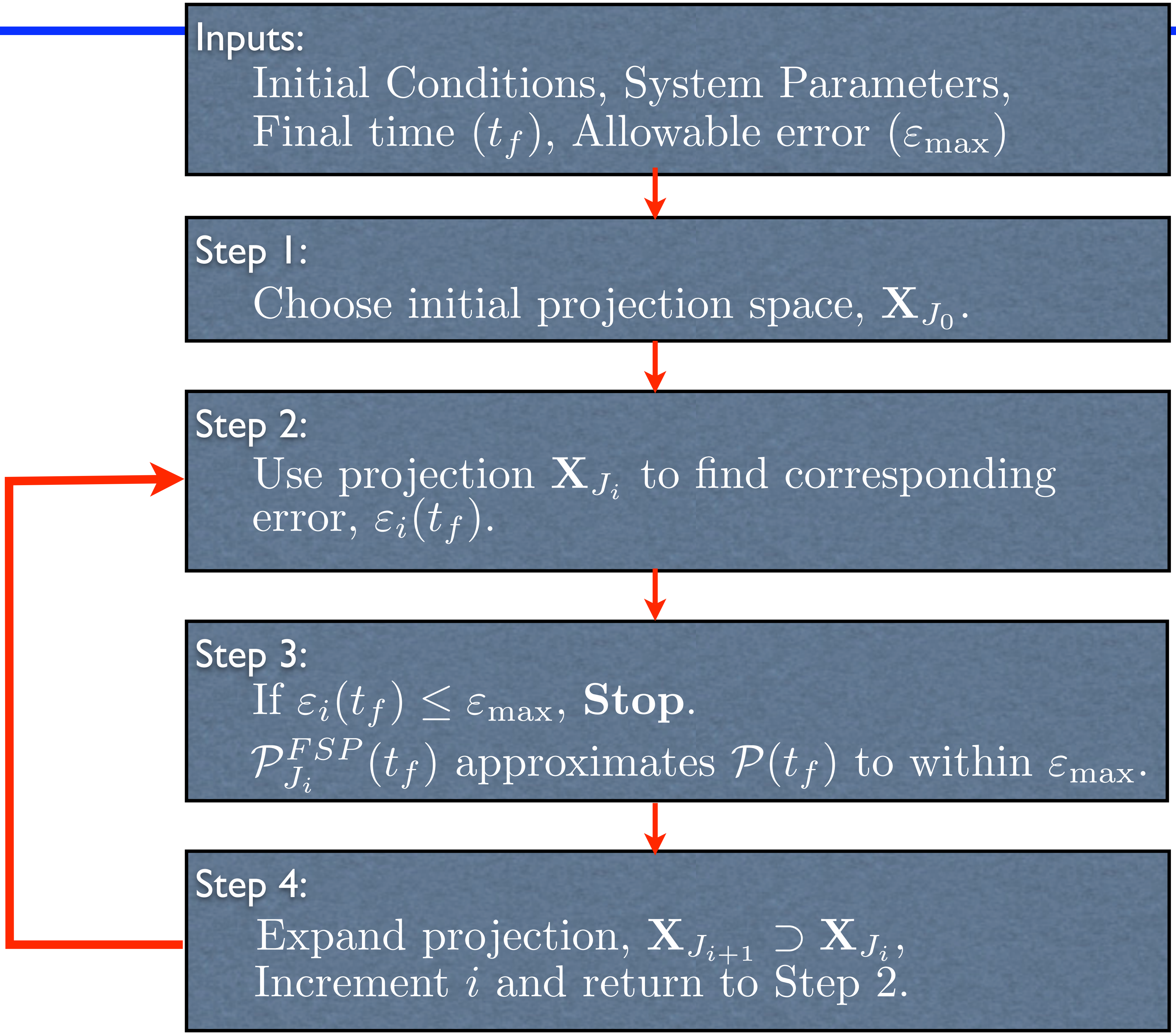
The FSP Theorem

(Munsky/Khammash JCP ‘06)

$\mathbf{P}_J(t) \geq \mathbf{P}_J^{FSP}(t)$ and

$$\left\| \begin{bmatrix} \mathbf{P}_J(t) \\ \mathbf{P}_{J'} \end{bmatrix} - \begin{bmatrix} \mathbf{P}_J^{FSP}(t) \\ \mathbf{0} \end{bmatrix} \right\|_1 = \varepsilon(t)$$

The Finite State Projection Algorithm



Inputs:

Initial Conditions, System Parameters, Final time (t_f), Allowable error ($\varepsilon_{\max}$)

Step 1:

Choose initial projection space, $\mathbf{X}_{J_0}$.

Step 2:

Use projection $\mathbf{X}_{J_i}$ to find corresponding error, $\varepsilon_i(t_f)$.

Step 3:

If $\varepsilon_i(t_f) \leq \varepsilon_{\max}$, **Stop**.

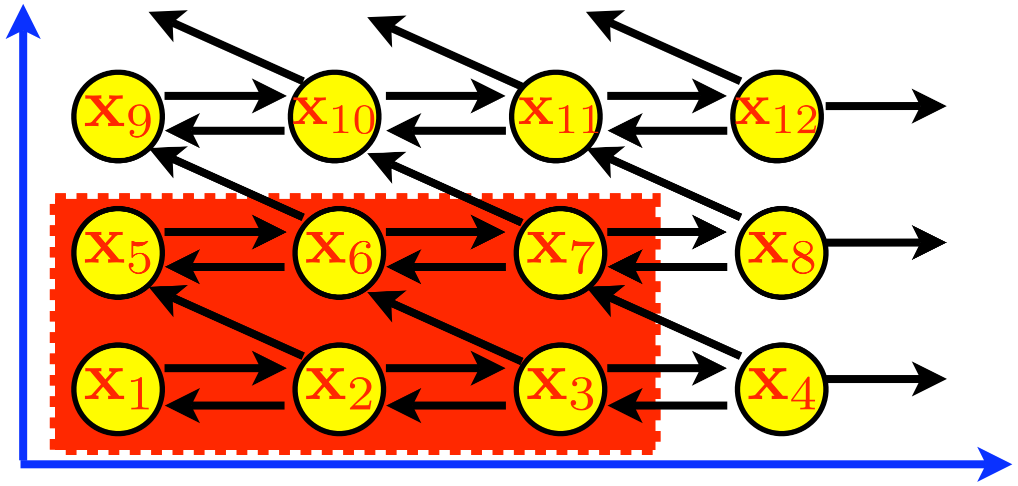
$\mathcal{P}^{FSP}_{J_i}(t_f)$ approximates $\mathcal{P}(t_f)$ to within $\varepsilon_{\max}$.

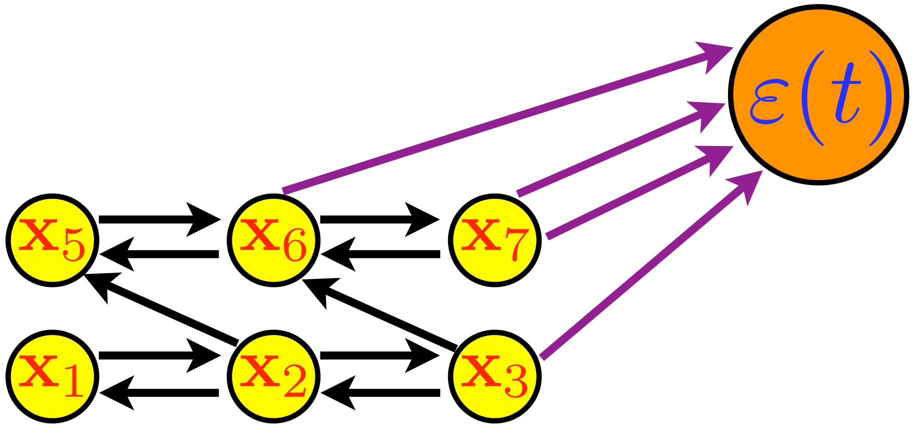
Step 4:

Expand projection, $\mathbf{X}_{J_{i+1}} \supset \mathbf{X}_{J_i}$,

Increment i and return to Step 2.

The “error” sink of the FSP to get exit times.

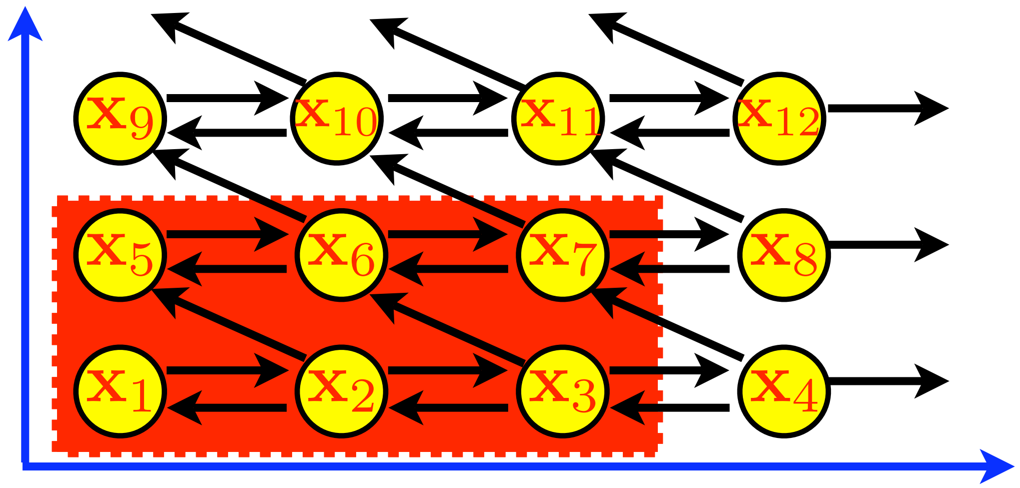


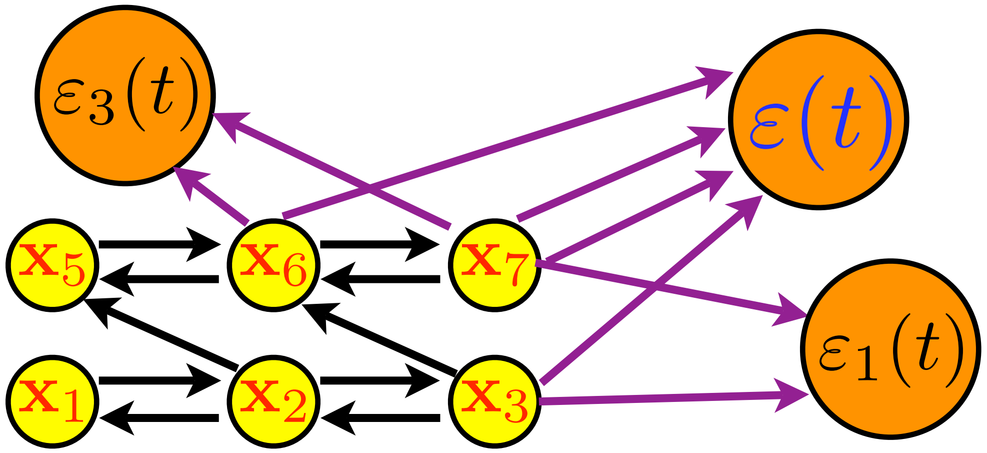


- In the original FSP, $\varepsilon(t)$ is the amount of the probability measure that exits the projection region $\mathbf{X}_J$.
- Median exit time: $t_{50} = t, \text{ s.t. } \varepsilon(t) = 0.5$
- In this form $\varepsilon(t)$ gives information as to when the system exits $\mathbf{X}_J$, but not how.

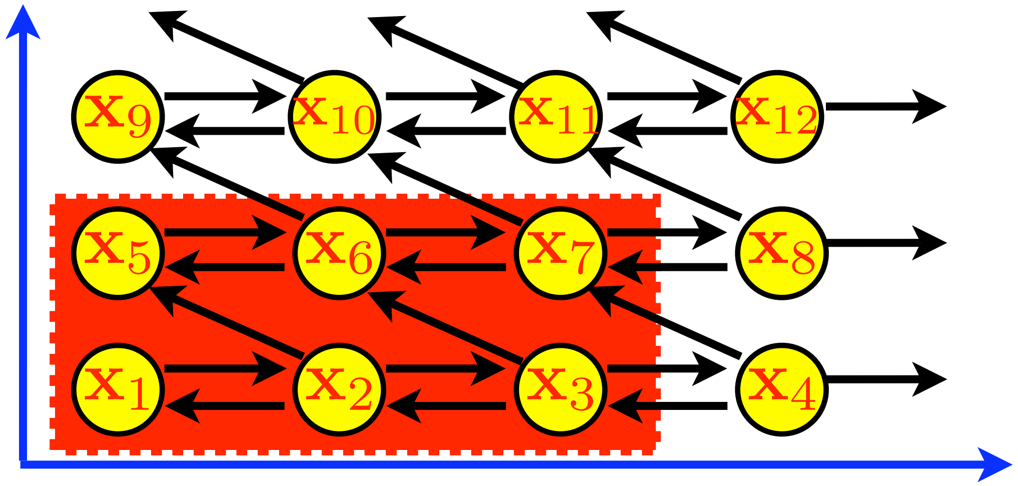
Multiple FSP sinks to get exit directions.

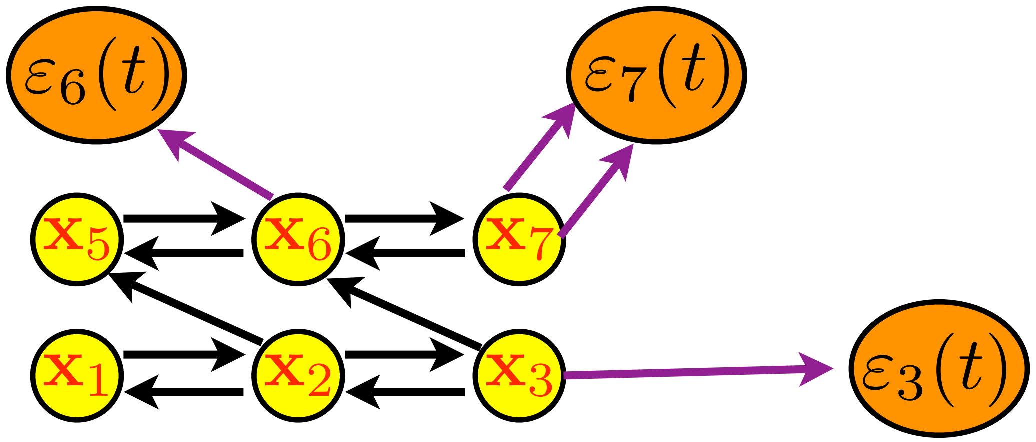
- By using multiple sinks, one can determine *how* the probability measure exits $\mathbf{X}_J$.





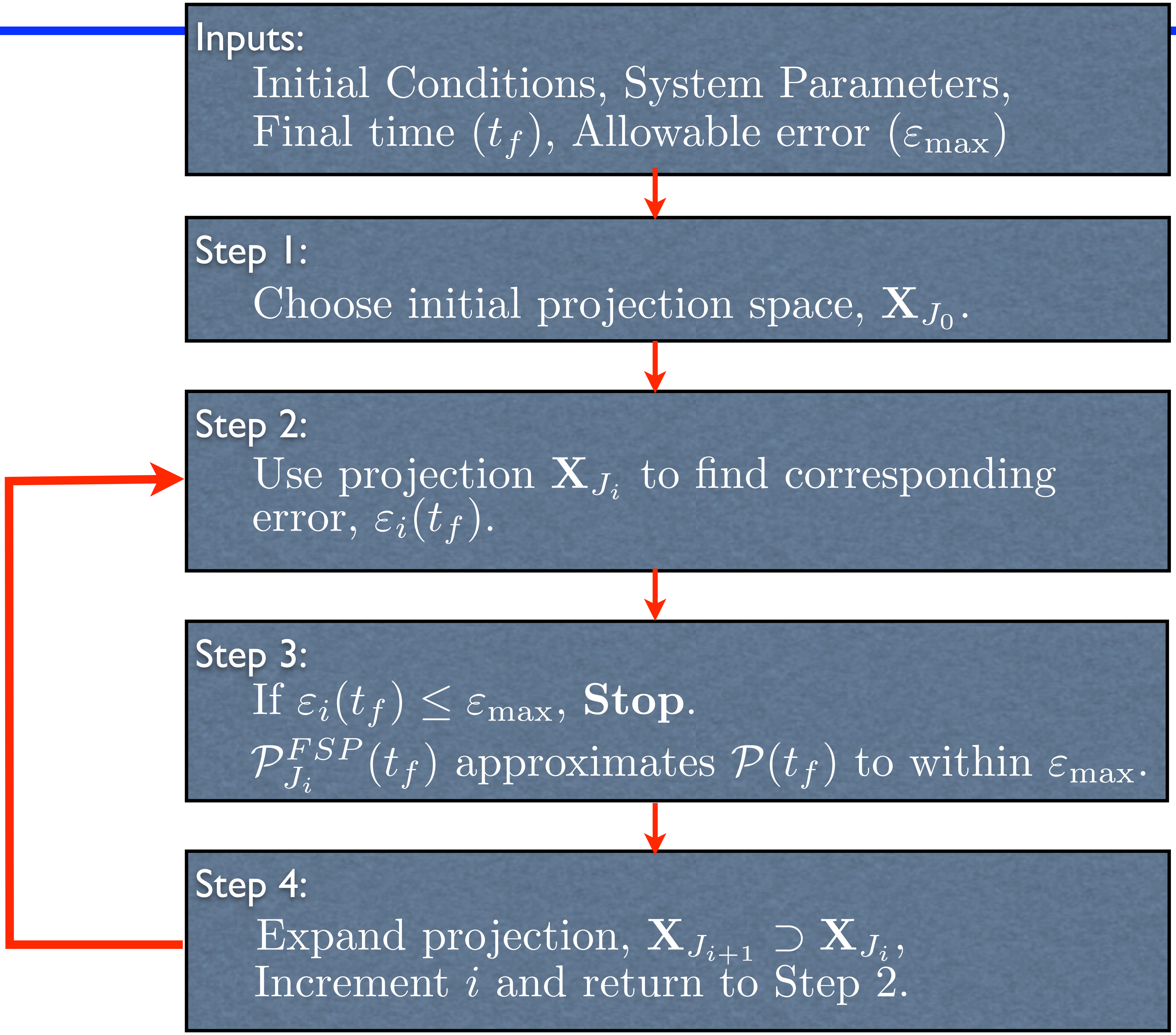
Which Reaction Leaves $\mathbf{X}_J$?





From which state?

The Finite State Projection Algorithm



Inputs:

Initial Conditions, System Parameters, Final time (t_f), Allowable error ($\varepsilon_{\max}$)

Step 1:

Choose initial projection space, $\mathbf{X}_{J_0}$.

Step 2:

Use projection $\mathbf{X}_{J_i}$ to find corresponding error, $\varepsilon_i(t_f)$.

Step 3:

If $\varepsilon_i(t_f) \leq \varepsilon_{\max}$, **Stop**.

$\mathcal{P}_{J_i}^{FSP}(t_f)$ approximates $\mathcal{P}(t_f)$ to within $\varepsilon_{\max}$.

Step 4:

Expand projection, $\mathbf{X}_{J_{i+1}} \supset \mathbf{X}_{J_i}$,

Increment i and return to Step 2.

Advantages of the FSP.

- Deterministic.
  - ★ Every run of the FSP yields the same result.
  - ★ Enables easier comparisons of different systems (sensitivity analysis).
- Provides accuracy guarantees.
  - ★ Can be made as precise as required.
  - ★ Allows for analysis of rare events.
- Does not depend upon initial conditions.
- Is open to many subsequent model reductions.

Limitations

- Numerical stiffness may lead to computational inefficiency.
- Systems may become very large as distributions cover large regions of the configuration space.
  - ★ Compact distributions may drift over time.
  - ★ Dilute distributions may spread over large regions.
  - ★ Dimension grows exponentially with the number of species.
- For these problems, the original FSP may not suffice,
- BUT, with additional model reductions and systematic techniques, many of these problems may be alleviated.

Outline

☑ Finite State Projection (FSP)

● Reductions to the FSP

★ Aggregating unobservable states

Munsky/Khammash, *CDC*, 2006

★ Time interval discretization

★ Slow manifold projection

★ Coarse meshes for the CME

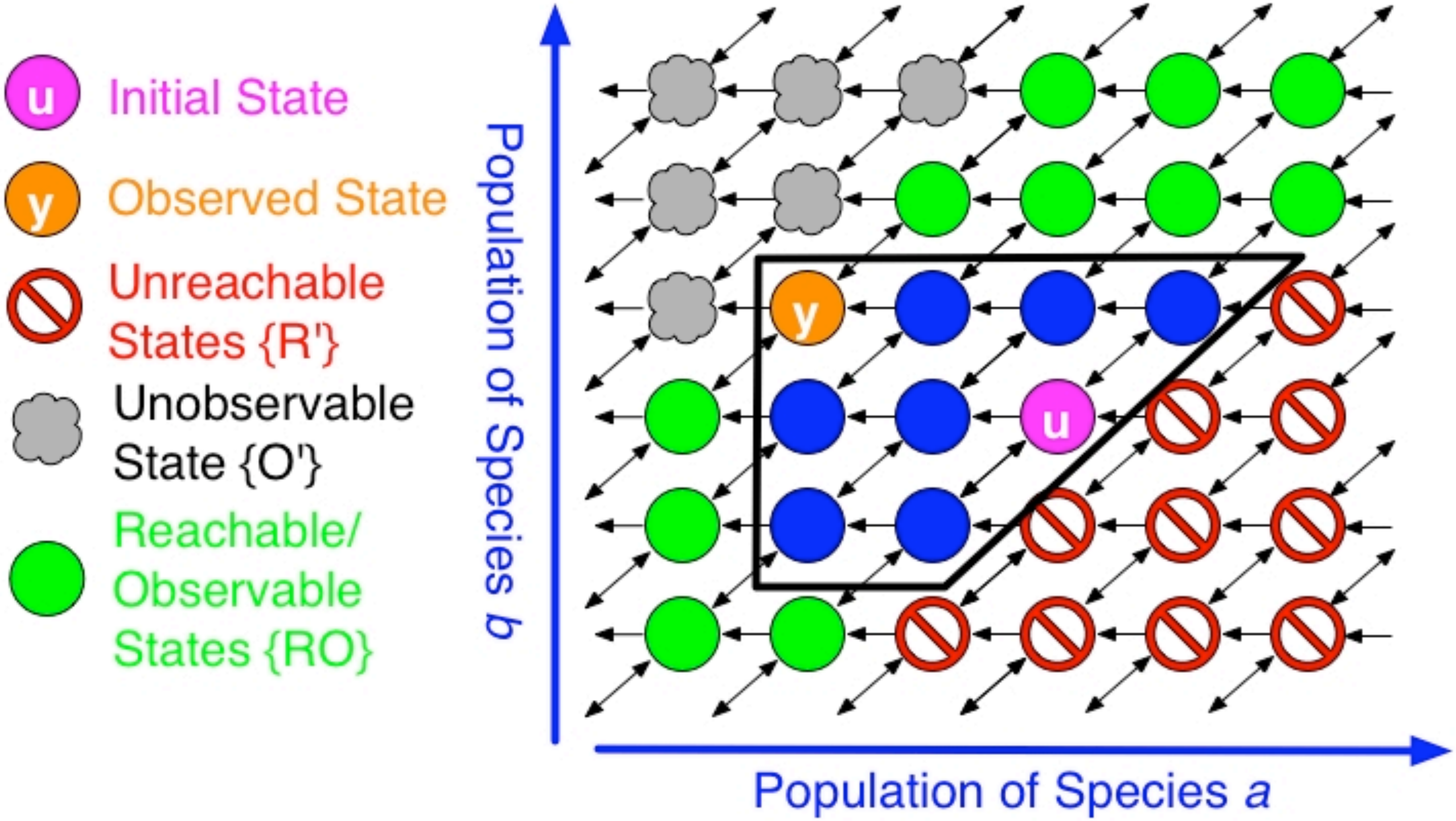
Using Input & Output relations for model reduction.

Brian Munsky

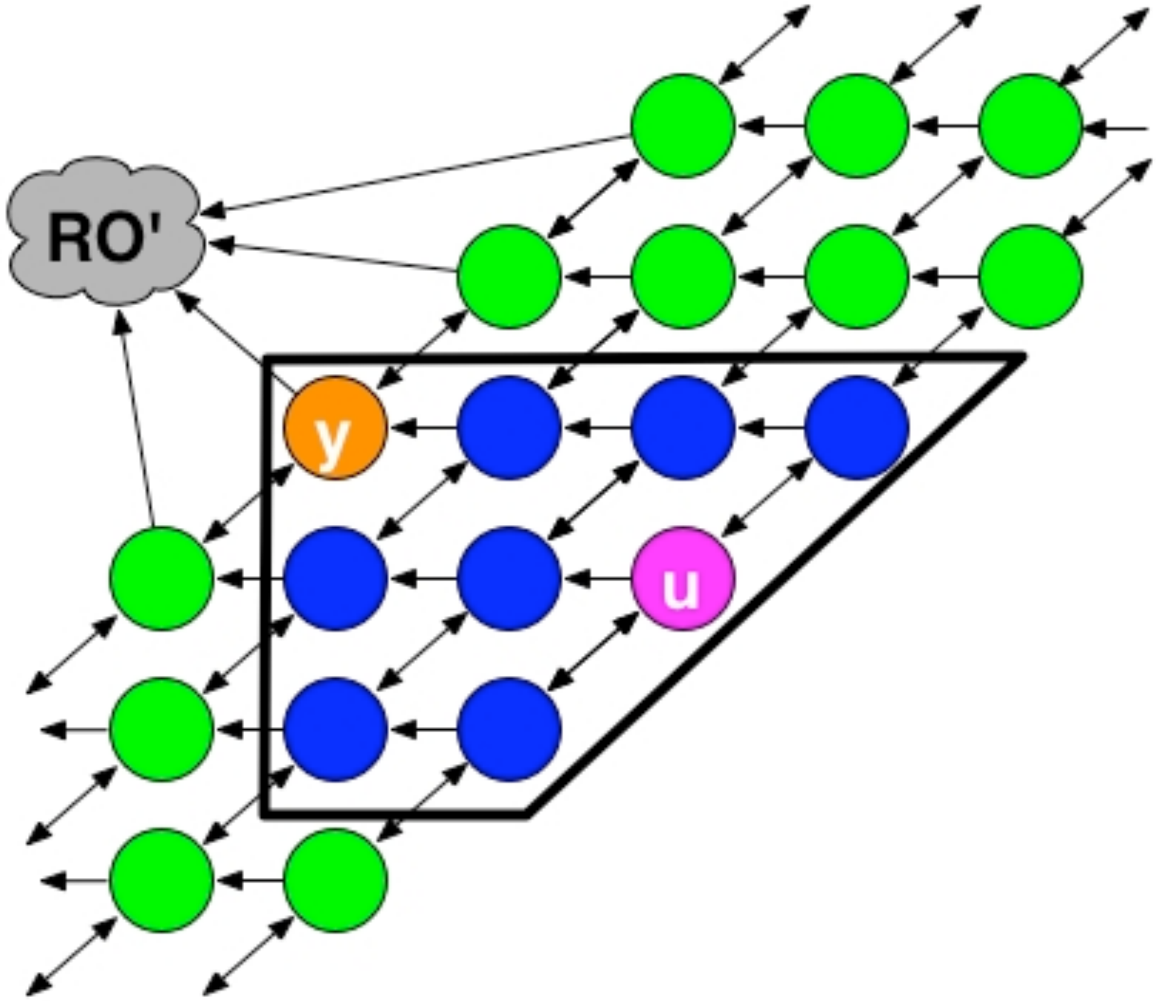
- Often one is not interested in the entire probability distribution.
- Instead one may wish only to estimate:
  - ★ a statistical summary of the distribution, e.g.
    - ✦ means, variances, or higher moments
  - ★ probability of certain traits:
    - ✦ switch rate, extinction, specific trajectories, etc…
- In each of these cases, one can define an output **y**(t):

$$\dot{\mathbf{P}}(t) = \mathbf{AP}(t)$$
$$\mathbf{y}(t) = \mathbf{CP}(t)$$

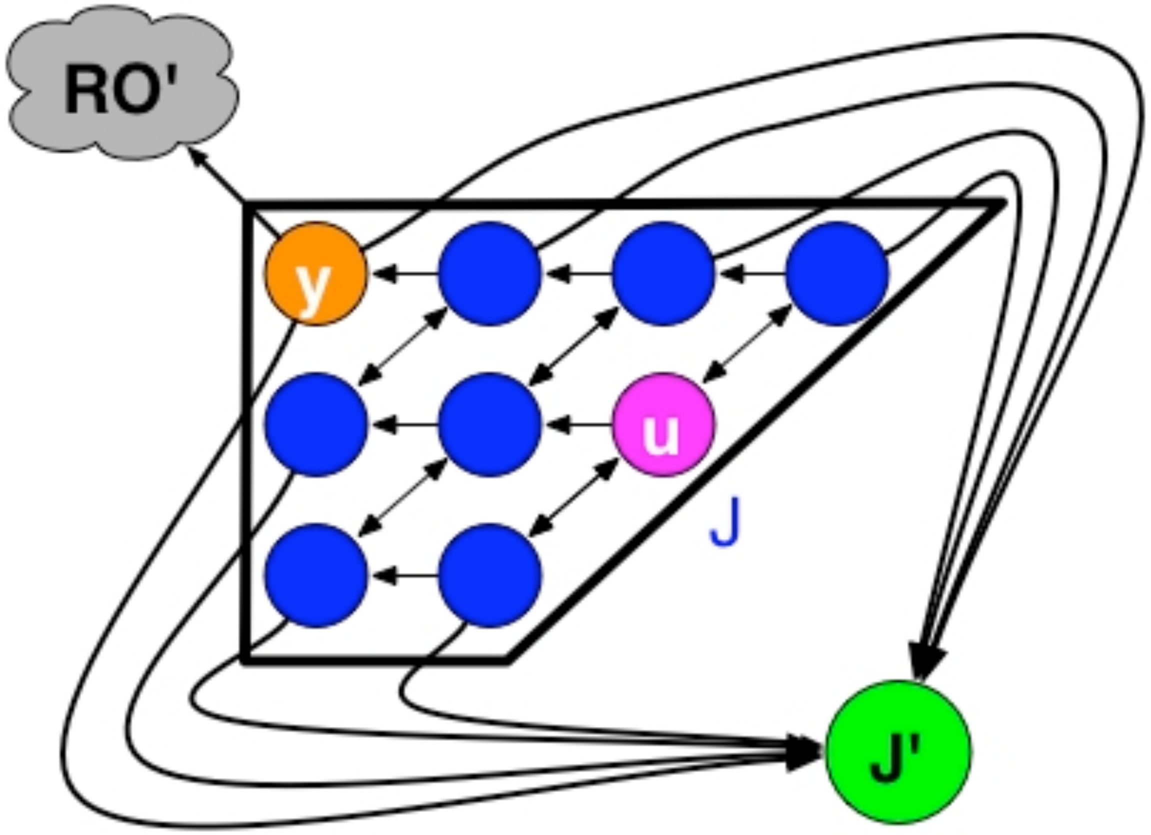
Begin with a Full Integer Lattice Description of the System States.


u
Initial State
y
Observed State
Unreachable States {R'}
Unobservable State {O'}
Reachable/ Observable States {RO}
Population of Species *b*
Population of Species *a*
y
u

Remove Unreachable States and Aggregate the Observable States.



Project the Reachable/Observable States onto a Finite Subspace.



We now have a solvable approximation, for which the FSP gives bounds on the approximation's accuracy.

Outline

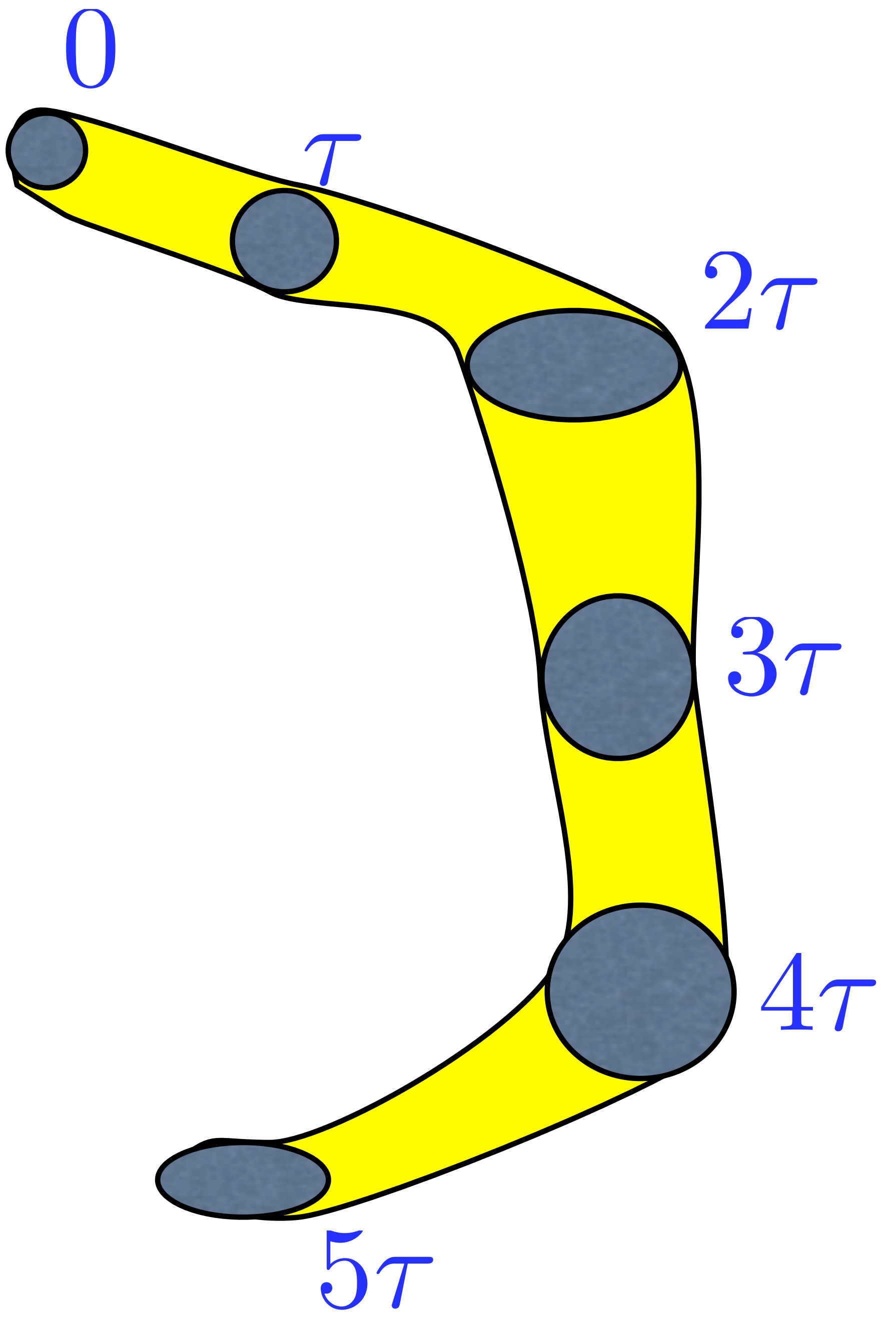
- ☑ Introduction
- ☑ Monte Carlo Solution Schemes
- ☑ Finite State Projection (FSP)
- ● Reductions to the FSP
  - ★ Minimal Realizations
  - ★ Time interval discretization

    Munsky and Khammash, *J. Comp. Phys.*, 2007

    Burrage et al, *A.A. Markov 150th Anniv. Meeting*, 2006
  - ★ Slow manifold projection
  - ★ Coarse meshes for the CME

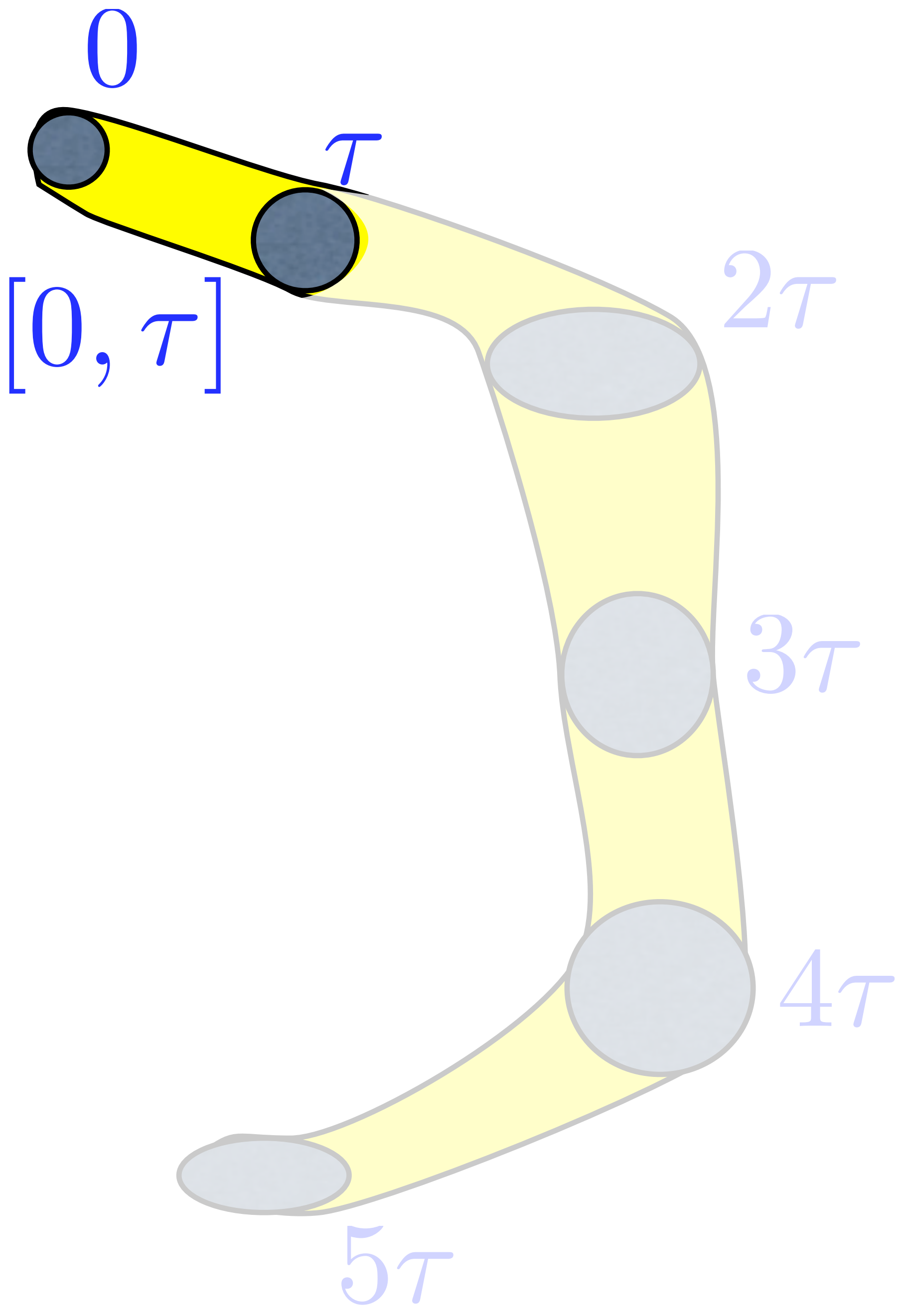
Time Interval Discretization for the FSP

- ★ For many systems, the distribution may drift over time.
- ★ At any one time, the distribution may have a limited support, but...
- ★ The FSP solution must include all intermediate configurations.
- ★ This may lead to an exorbitantly large system of ODEs.



Time Interval Discretization for the FSP

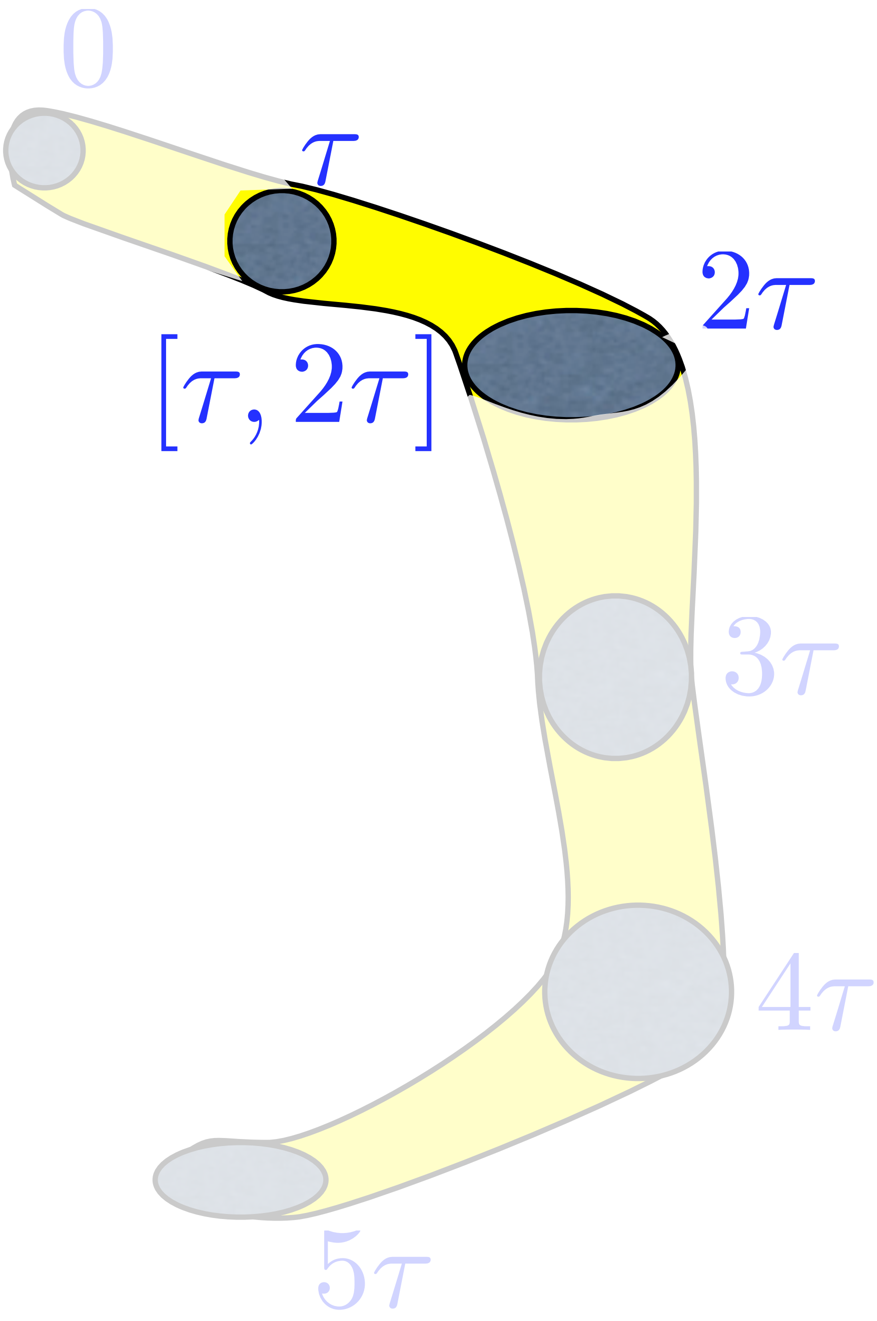
- ★ Instead:
  - ✳ Discretize the time interval into smaller steps and solve a separate projection for each interval.



Time Interval Discretization for the FSP

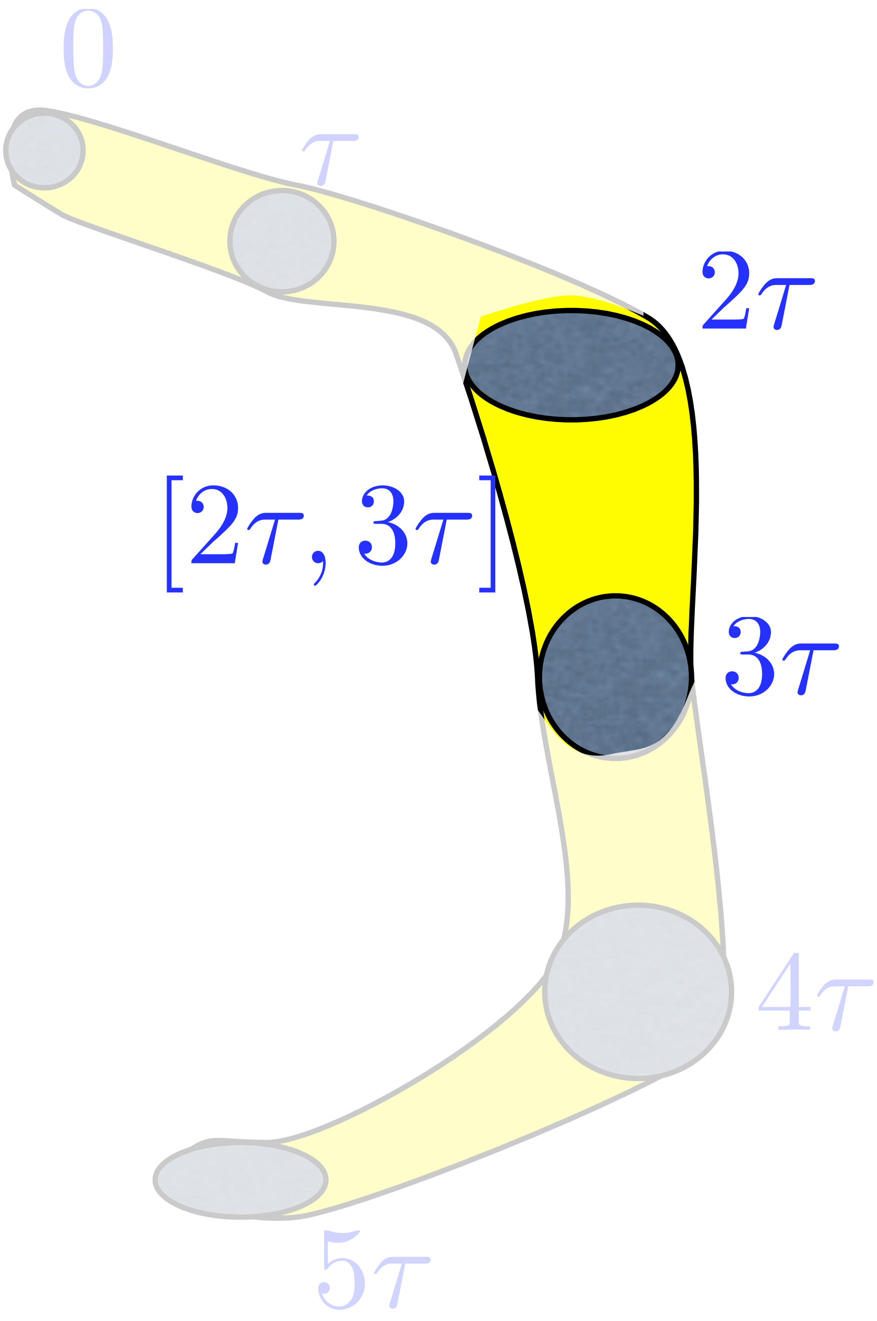
Brian Munsky

- ★ Instead:
  - ✳ Discretize the time interval into smaller steps and solve a separate projection for each interval.



Time Interval Discretization for the FSP

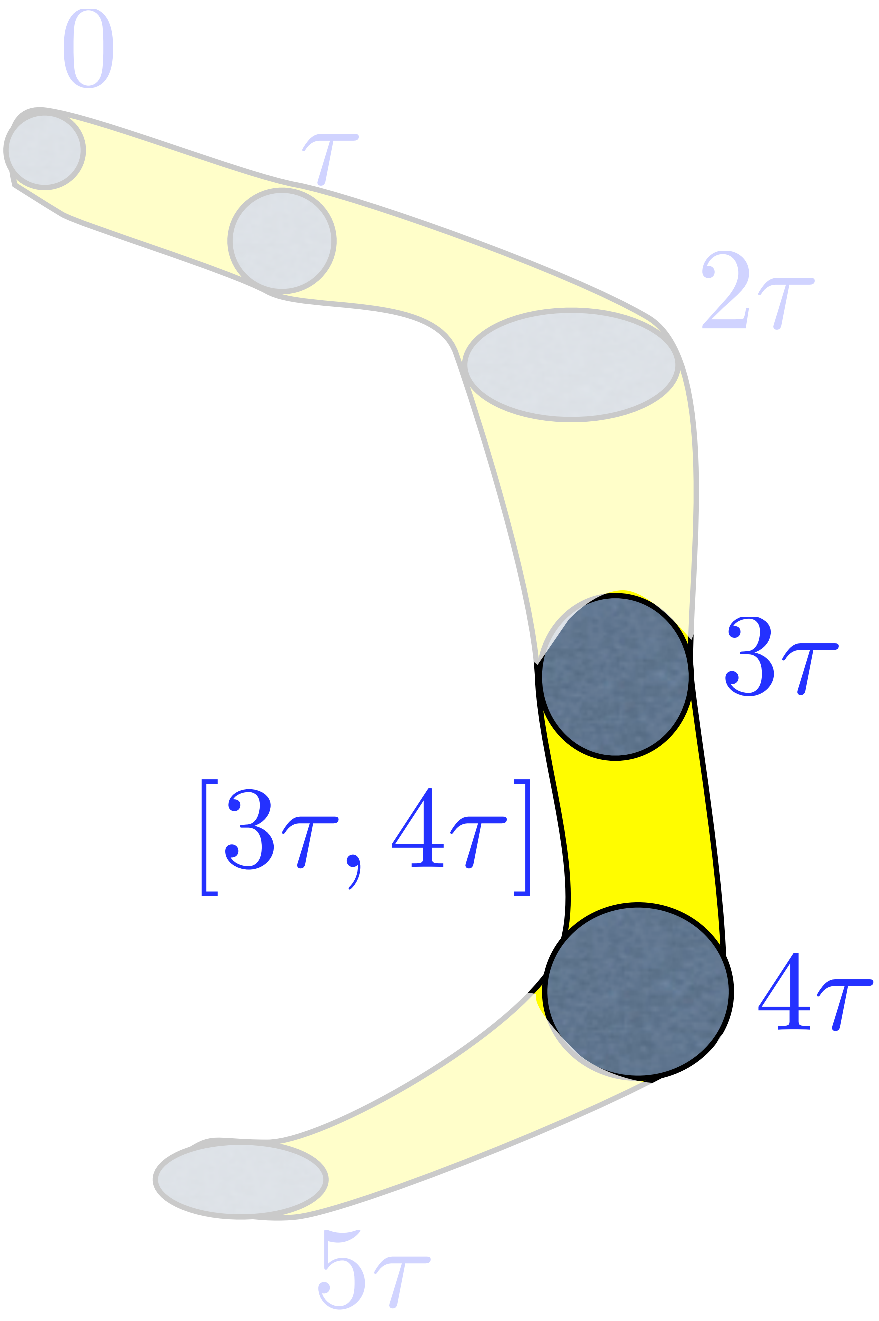
- ★ Instead:
  - ✳ Discretize the time interval into smaller steps and solve a separate projection for each interval.



Time Interval Discretization for the FSP

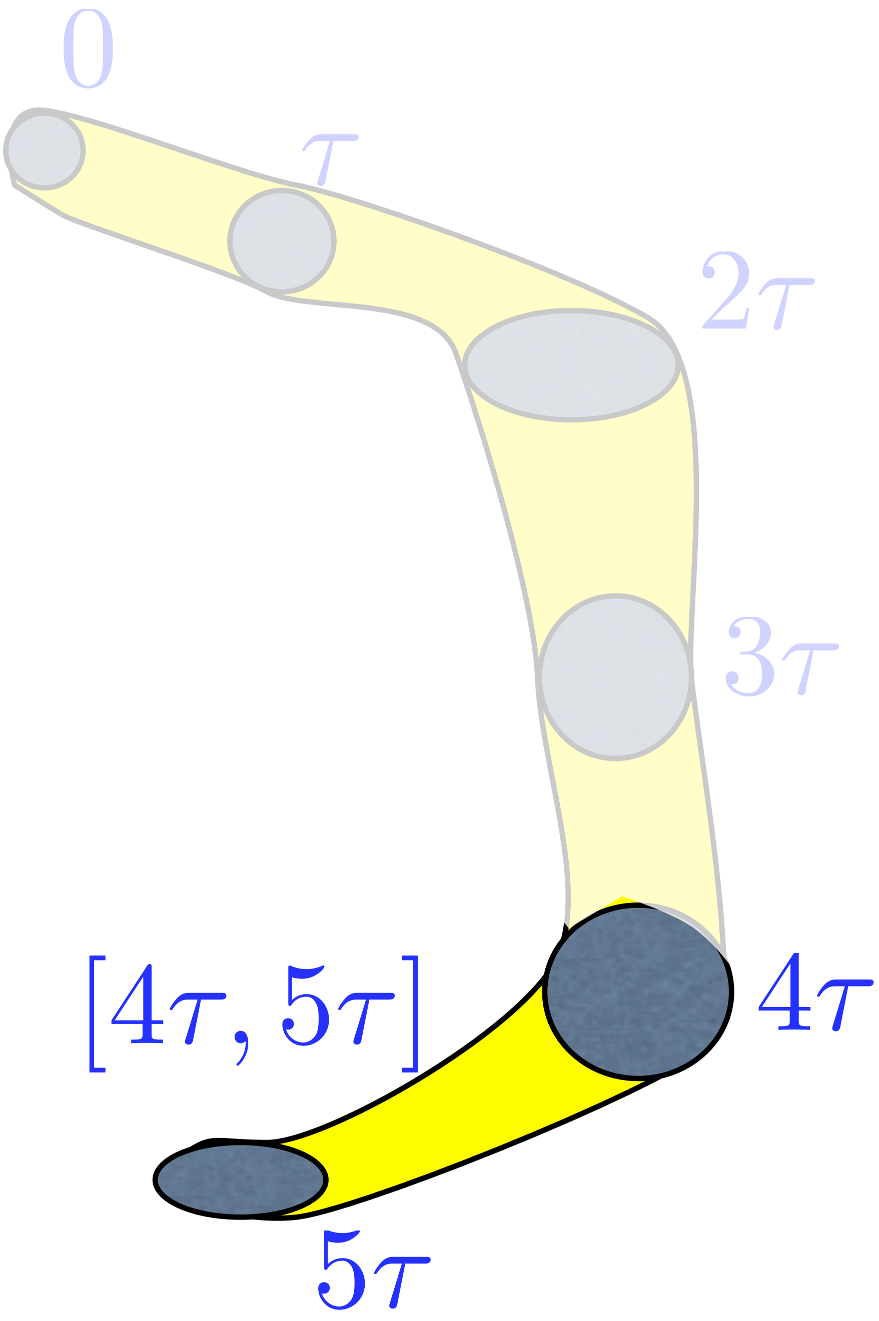
Brian Munsky

- ★ Instead:
  - ✳ Discretize the time interval into smaller steps and solve a separate projection for each interval.



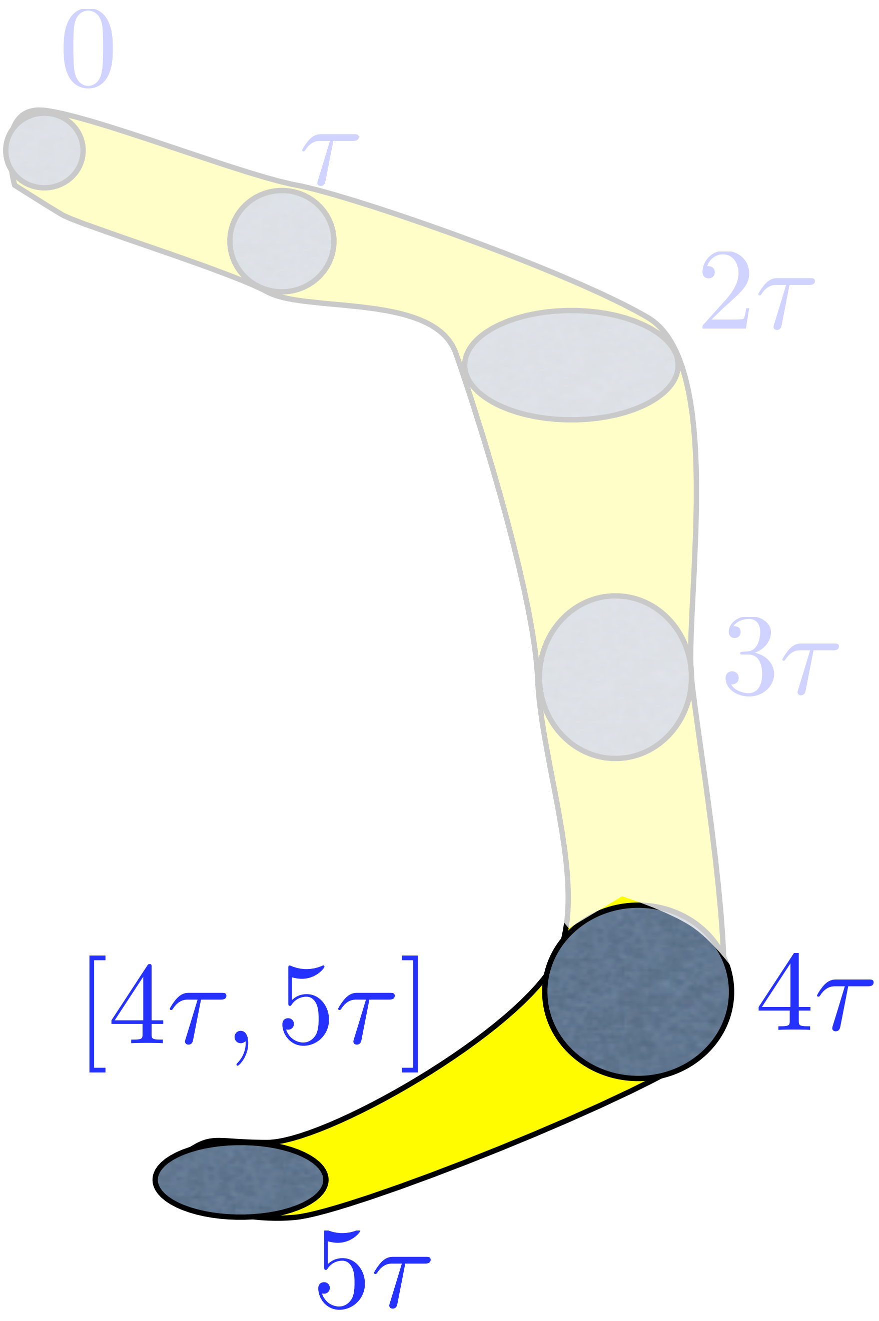
Time Interval Discretization for the FSP

- ★ Instead:
  - ✳ Discretize the time interval into smaller steps and solve a separate projection for each interval.



Time Interval Discretization for the FSP

- ★ Solving a few smaller systems can be much easier than solving a single large system.
- ★ Control the error at each step to obtain a guaranteed final error.
- ★ Caching and reusing information from one step to the next may further reduce effort.



Outline

- ☑ Introduction
- ☑ Monte Carlo Solution Schemes
- ☑ Finite State Projection (FSP)
- ● Reductions to the FSP
  - ★ Minimal Realizations
  - ★ Time interval discretization
  - ★ Slow manifold projection

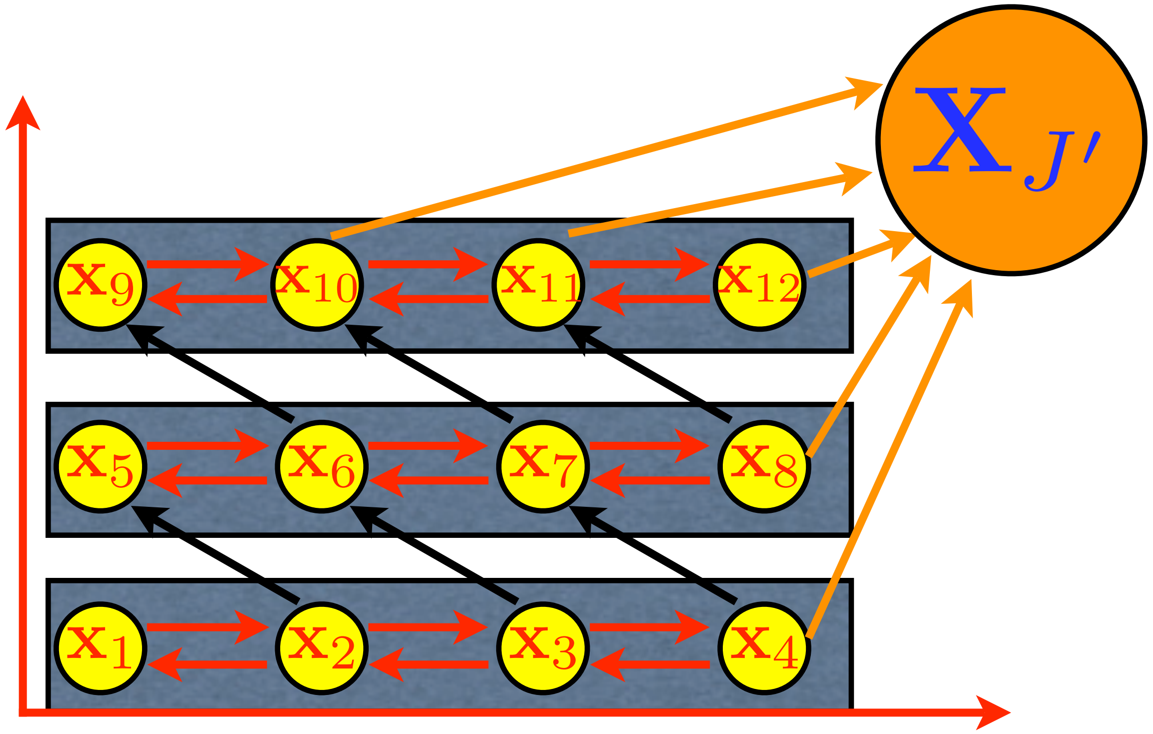
    Peles/Munsky/Khammash, *JCP,* 2006
  - ★ Coarse meshes for the CME

Perturbation Theory and the FSP

- Some reactions occur faster and more frequently than others.
- This can result in a separation of time-scales in the CME.
  - Disadvantages: Often results in numerical stiffness and increased computational complexity.
  - Advantage: May be able to apply perturbation theory to reduce computational effort.

Intuition (Time Scale Separation)

1. Begin with a finite state (projected) Markov process.
2. Group states connected by frequent reactions.



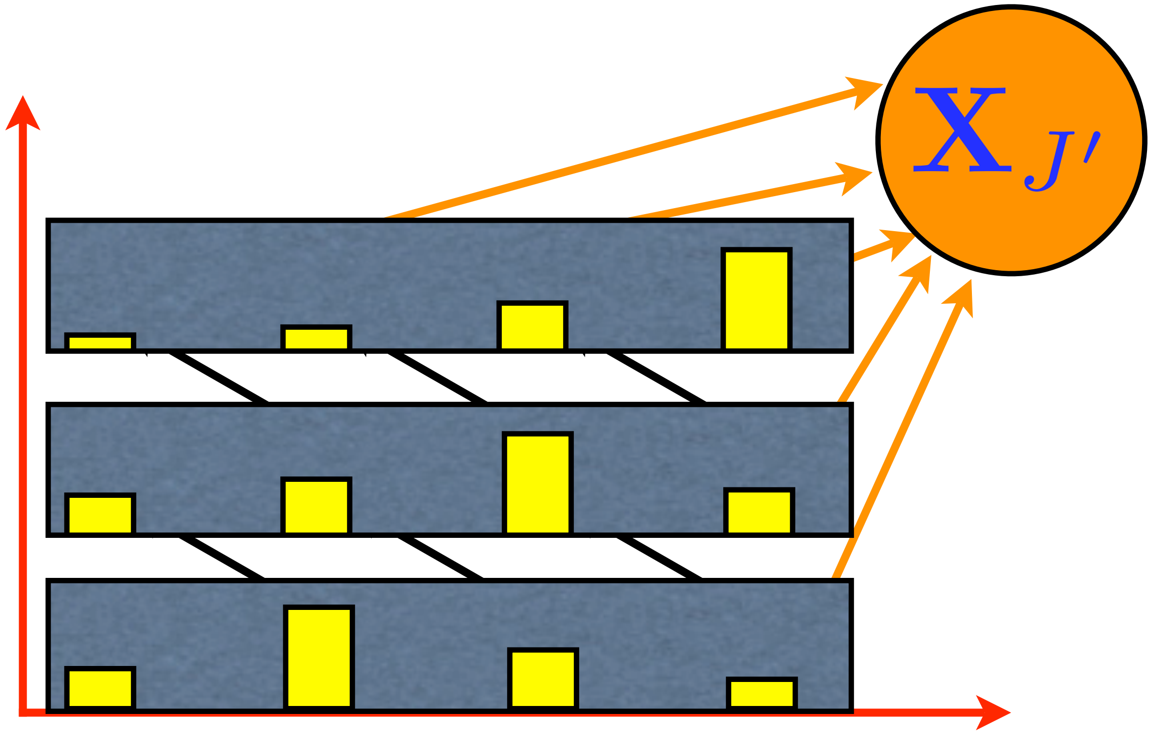
Red Arrows = Fast (Frequent) Reactions

Black Arrows = Slow (Rare) Reactions

Orange Arrows = (Rare) Transitions to Sink

Intuition (Time Scale Separation)

1. Begin with a finite state (projected) Markov process.
2. Group states connected by frequent reactions.
3. Find invariant distribution for each group.



Red Arrows = Fast (Frequent) Reactions

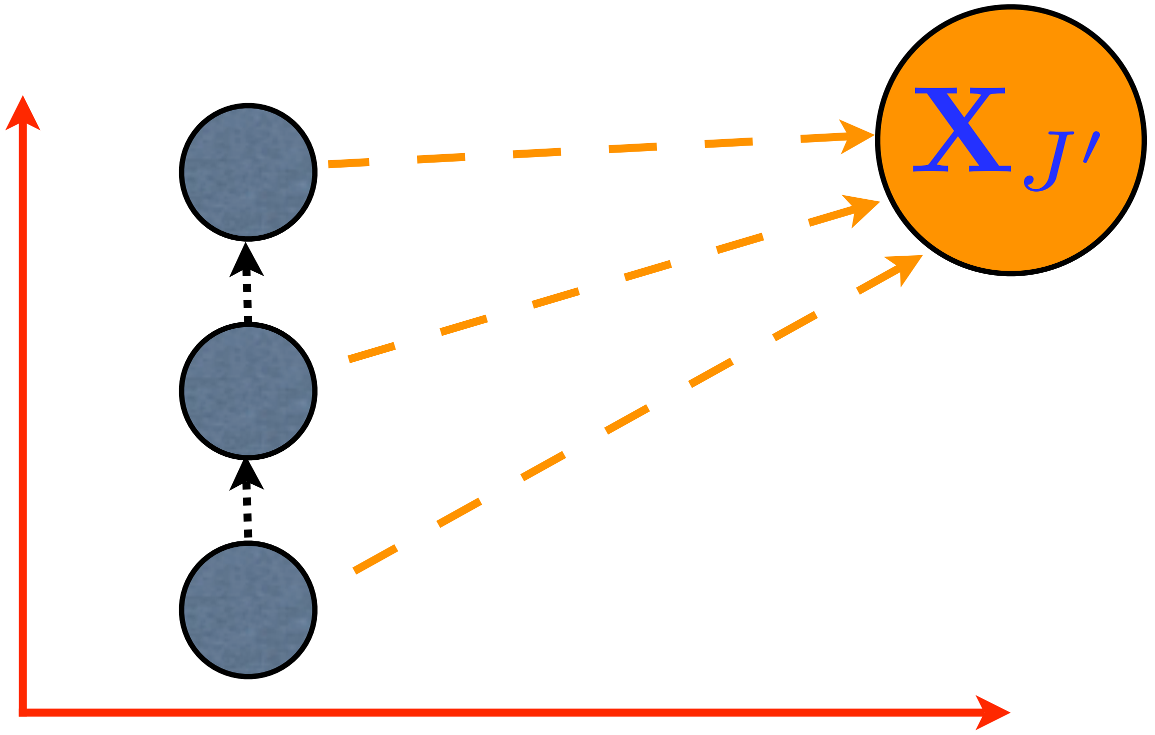
Black Arrows = Slow (Rare) Reactions

Orange Arrows = (Rare) Transitions to Sink

Intuition (Time Scale Separation)

1. Begin with a finite state (projected) Markov process.
2. Group states connected by frequent reactions.
3. Find invariant distribution for each group.
4. Average to find the rates of the slow reactions.

Reduced Markov Process



Dotted Black = Averaged Slow Reactions

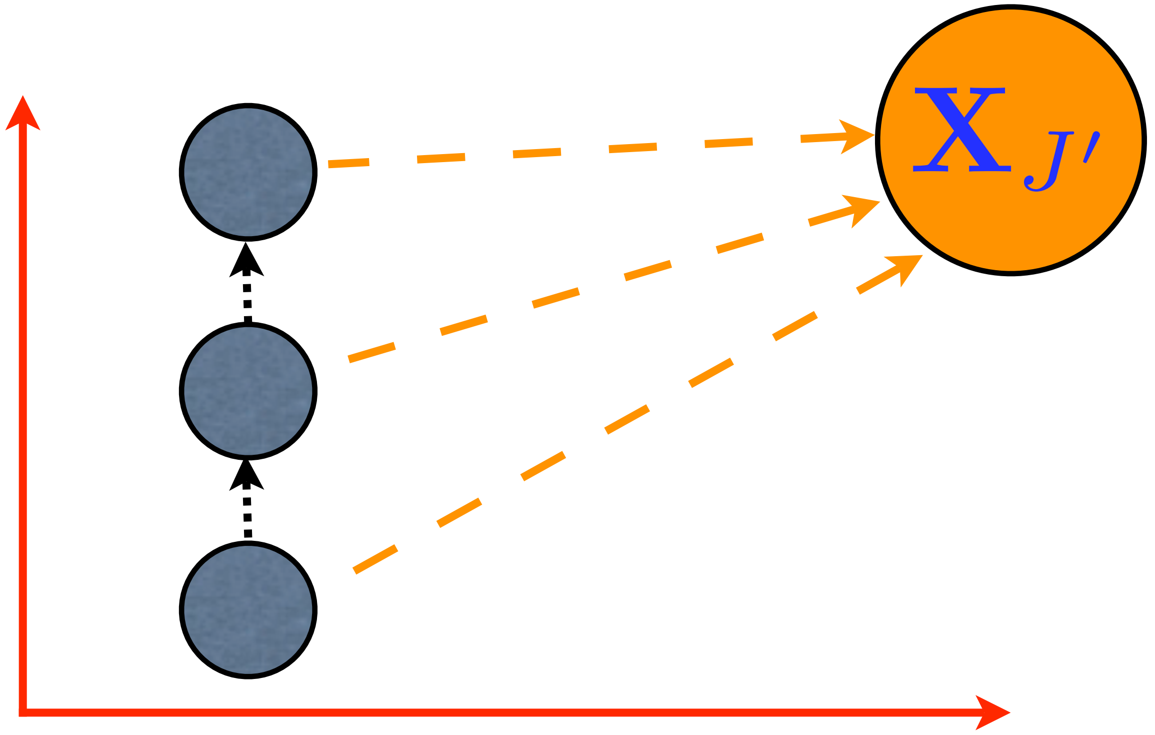
Dashed Orange = Averaged Transitions to Sink

Intuition (Time Scale Separation)

Brian Munsky

1. Begin with a finite state (projected) Markov process.
2. Group states connected by frequent reactions.
3. Find invariant distribution for each group.
4. Average to find the rates of the slow reactions.
5. Solve for the solution on the slow-manifold.
6. Lift solution to original coordinate system.

Reduced Markov Process



Dotted Black = Averaged Slow Reactions

Dashed Orange = Averaged Transitions to Sink

Outline

☑ Monte Carlo Solution Schemes

☑ Finite State Projection (FSP)

4. Reductions to the FSP

- ★ Minimal Realizations
- ★ Time interval discretization
- ★ Slow manifold projection
- ★ Coarse meshes for the CME

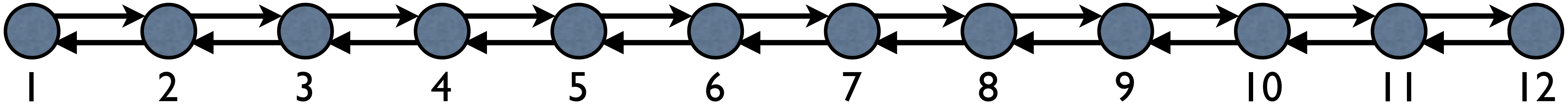
  Munsky/Khammash, *IEEE Trans*, 2008

Coarse mesh approximation of the CME

- Precision requirements may change for different regions of the configurations space.
  - ★ Small populations require great precision.
  - ★ High populations require far less precision.
- By choosing a good coarse approximation of the CME, we can take advantage of this.

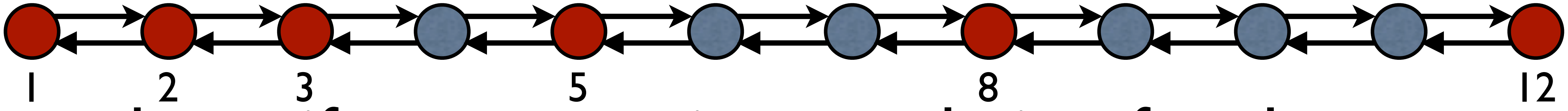
Coarse mesh approximation of the CME

Start with the full 1-dimensional Markov lattice.



$\dot{\mathbf{P}} = \mathbf{A}\cdot\mathbf{P}(t)$ Original CME

Choose a subset of mesh points.



and specify an approximate relation for the probability of the removed points: $\mathbf{P} \approx \mathbf{\Phi}\mathbf{q}(t)$

Solve the reduced system ODE: $\dot{\mathbf{q}} = \mathbf{\Phi}^{-L}\mathbf{A}\mathbf{\Phi}\mathbf{q}(t)$

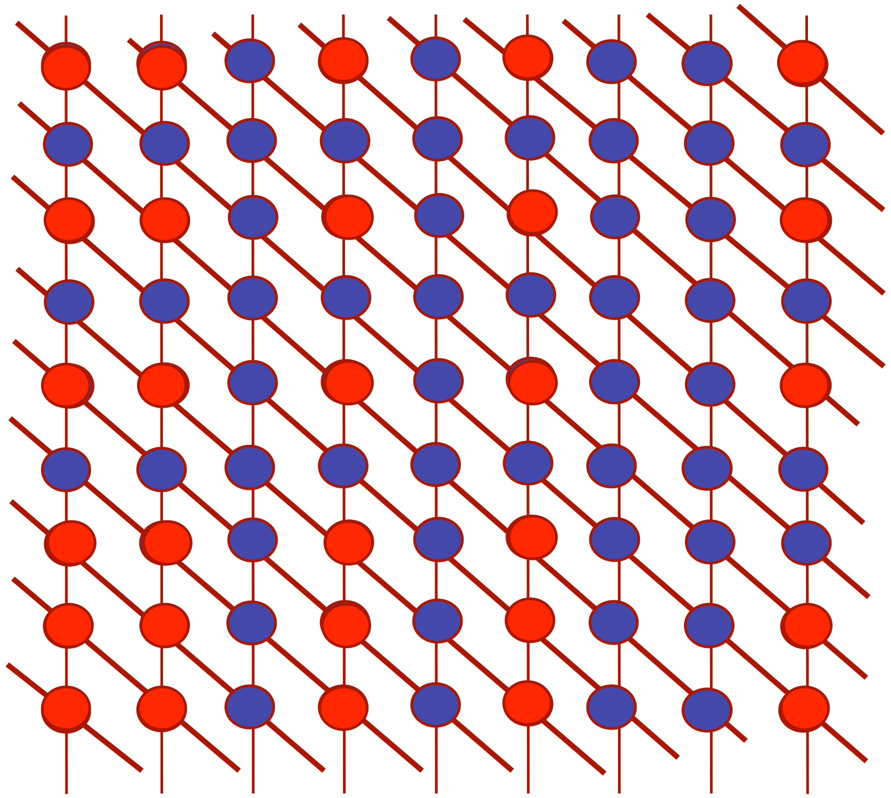
and lift back to the original system coordinates:

$$\mathbf{P}(t) \approx \mathbf{\Phi}\exp(\mathbf{\Phi}^{-L}\mathbf{A}\mathbf{\Phi}t)\mathbf{\Phi}^{-L}\mathbf{P}(0)$$

Coarse Mesh: Multiple-species problems.

For problems with many species, the method is the same.

1. Begin with original lattice.
2. Choose interpolation points.
3. Form interpolation (shape) function: $\mathbf{P}(t) \approx \mathbf{\Phi q}(t)$
4. Project system to find reduced system of ODEs:
   $\dot{\mathbf{q}}(t) = \mathbf{\Phi}^{-L}\mathbf{A\Phi q}(t)$
5. Solve reduced system.
6. Lift back to original coordinates.



Outline

- ☑ Monte Carlo Solution Schemes
- ☑ Finite State Projection (FSP)
- ☑ Reductions to the FSP
  - ● Example: Heat Shock.
  - ● Toggle Switch

The Heat Shock Mechanism

- To survive/compete in a changing environment, biology must quickly adapt to fluctuations in:
  - ★ Temperature, ph level, nutrient availability, etc...

- High temperature → proteins misfold.
- Heat-shock proteins are created to help fix or remove these misfolded proteins.

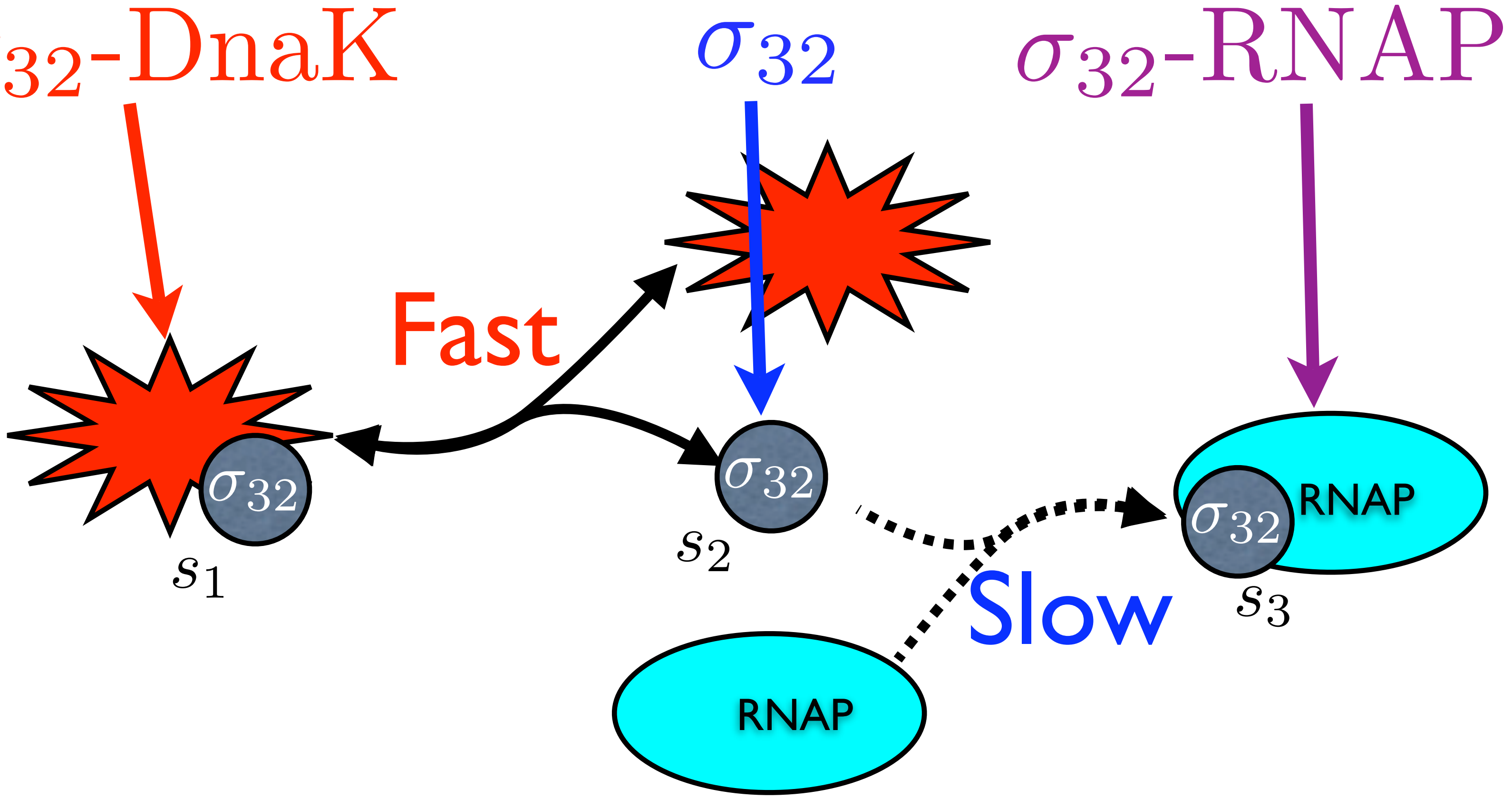
Toy Heat Shock Model in *E. coli*

Brian Munsky

3 forms for σ_{32}:

$$S_1 \underset{k_2}{\overset{k_1}{\rightleftarrows}} S_2$$

$$S_2 \xrightarrow{k_3} S_3$$

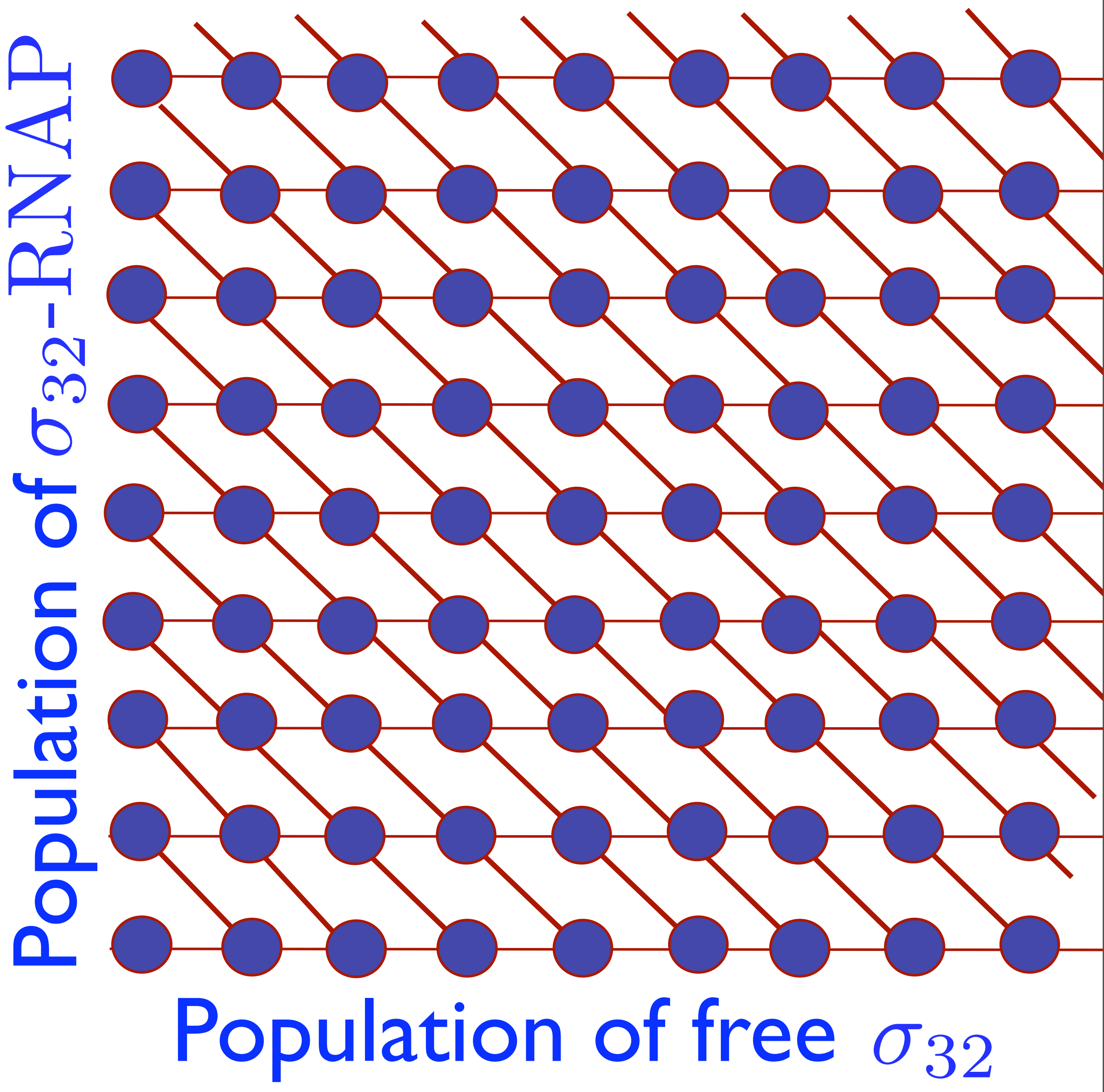


El Samad et al, *PNAS*, vol. 102, No. 8, 2005

Toy Heat Shock Model in *E. coli* (cont.)

Five Different FSP Solution Schemes:

1. Full FSP


Population of σ_{32}-RNAP
Population of free σ_{32}

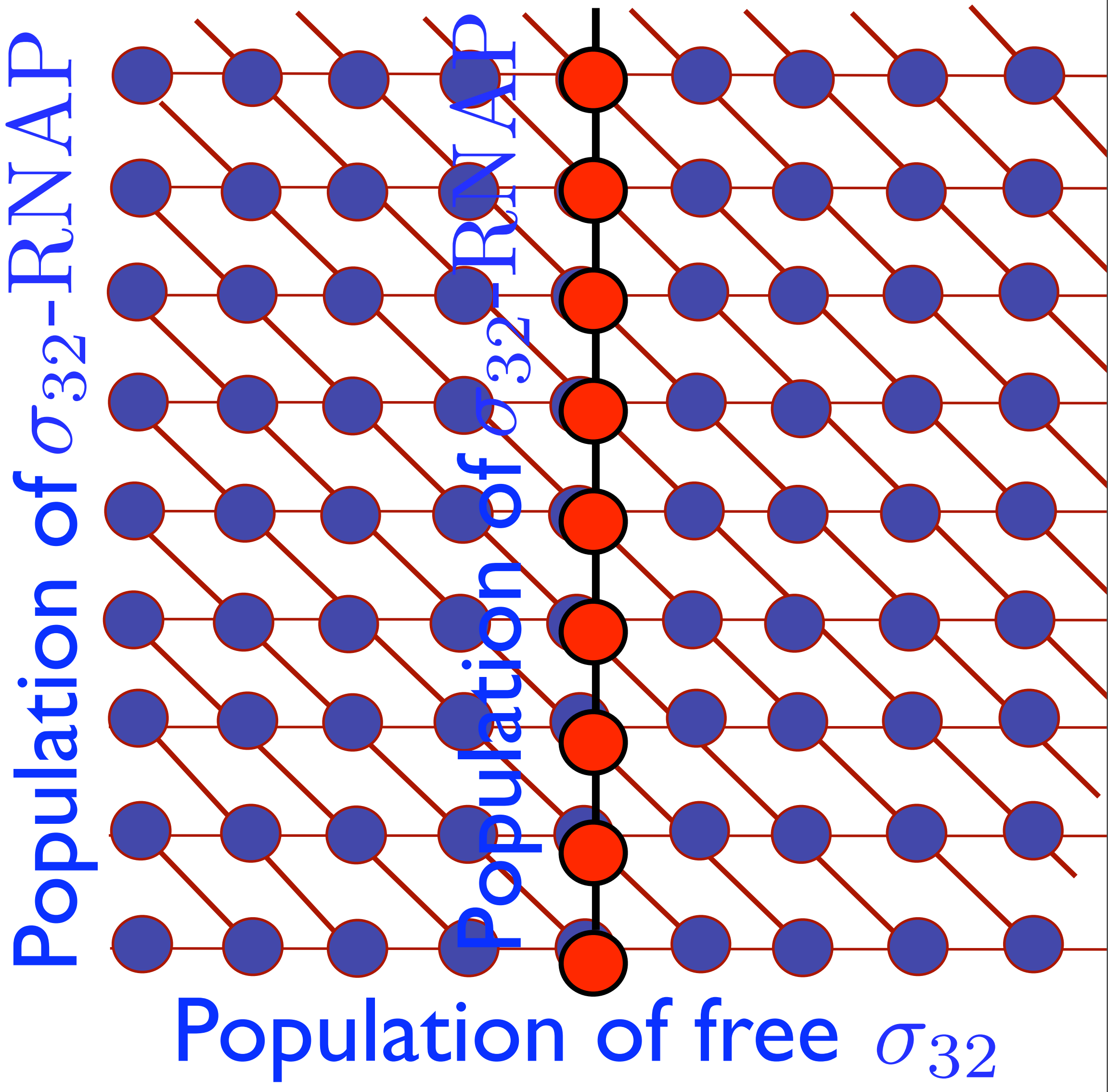
4459 ODEs

Toy Heat Shock Model in *E. coli* (cont.)

Brian Munsky

Five Different FSP Solution Schemes:

1. Full FSP
2. Slow manifold (FSP-SM)

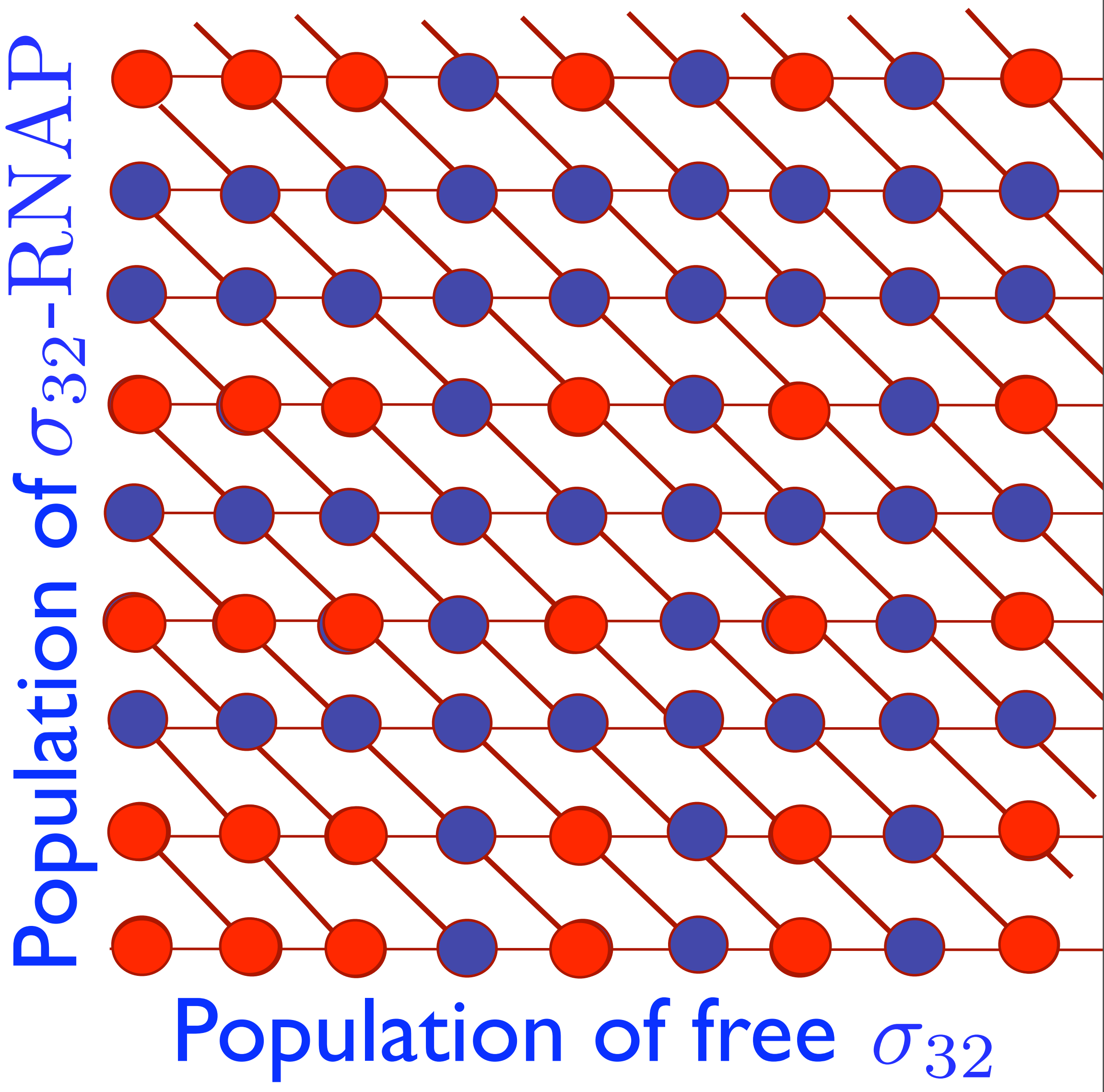

Population of σ_{32}-RNAP
Population of σ_{32}-RNAP
Population of free σ_{32}

343 ODEs

Toy Heat Shock Model in *E. coli* (cont.)

Five Different FSP Solution Schemes:

1. Full FSP
2. Slow manifold (FSP-SM)
3. Interpolated (FSP-I)

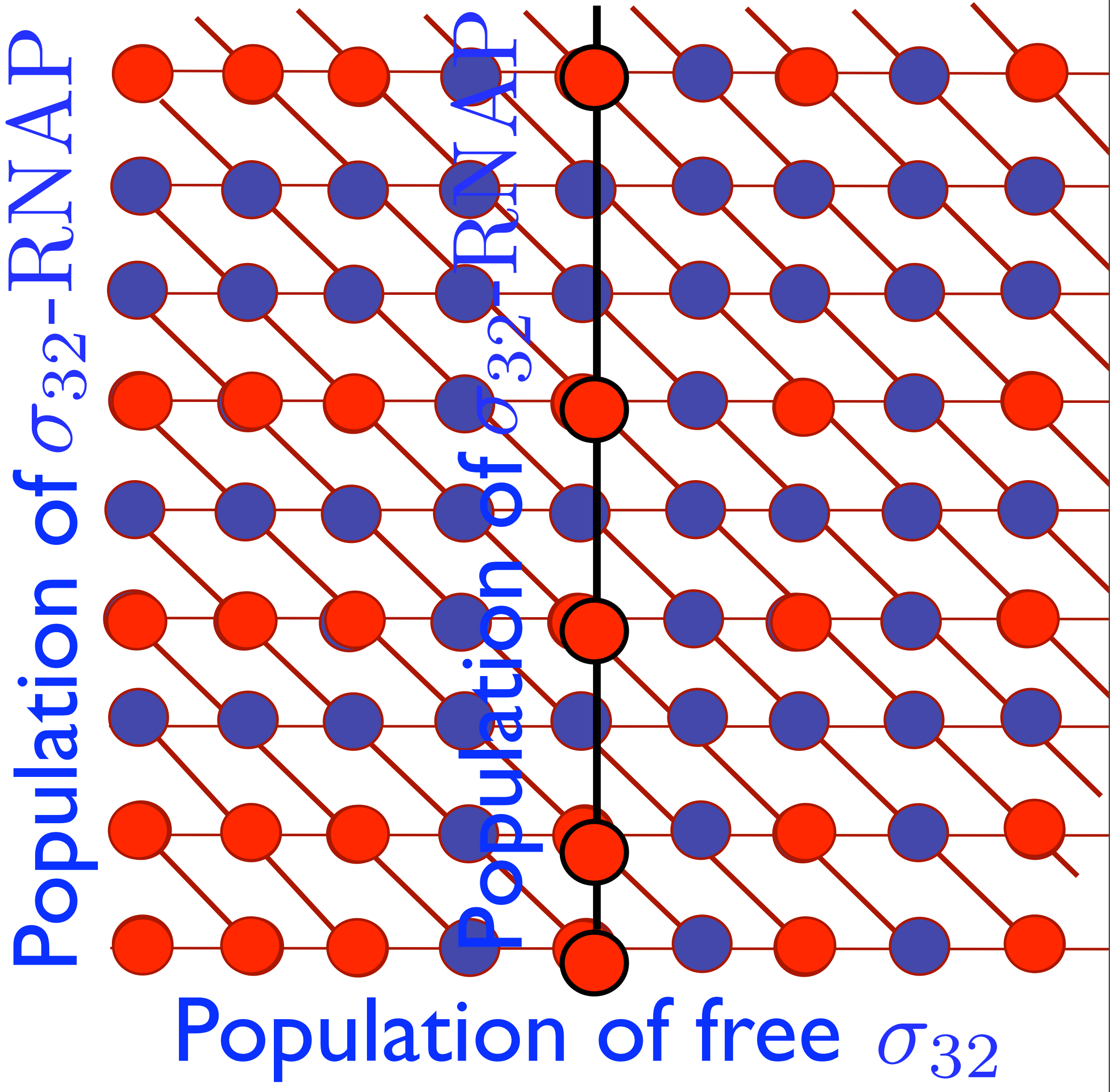

Population of σ_{32}-RNAP
Population of free σ_{32}

539 ODEs

Toy Heat Shock Model in *E. coli* (cont.)

Five Different FSP Solution Schemes:

1. Full FSP
2. Slow manifold (FSP-SM)
3. Interpolated (FSP-I)
4. Hybrid (FSP-SM/I)


Population of σ_{32}-RNAP
Population of σ_{32}-RNAP
Population of free σ_{32}

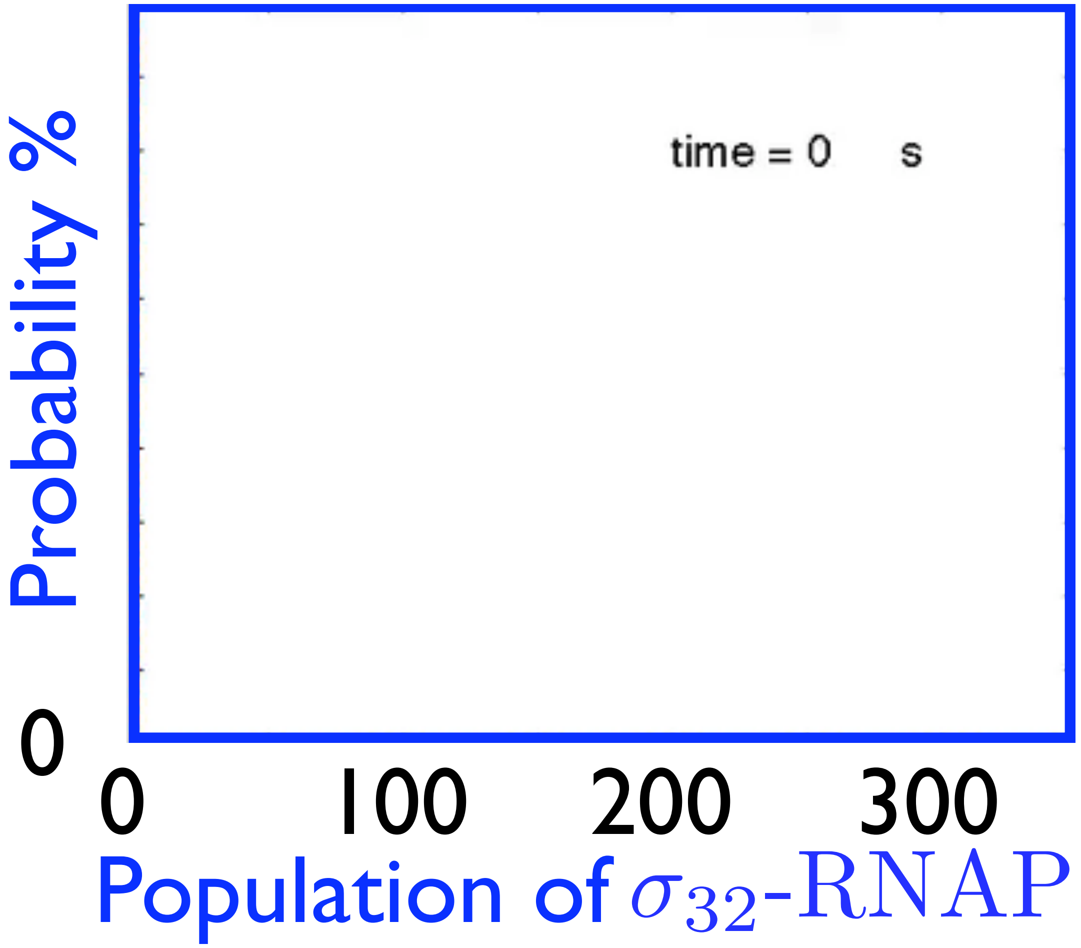
49 ODEs

Toy Heat Shock Model in *E. coli* (cont.)

Brian Munsky

Five Different FSP Solution Schemes:

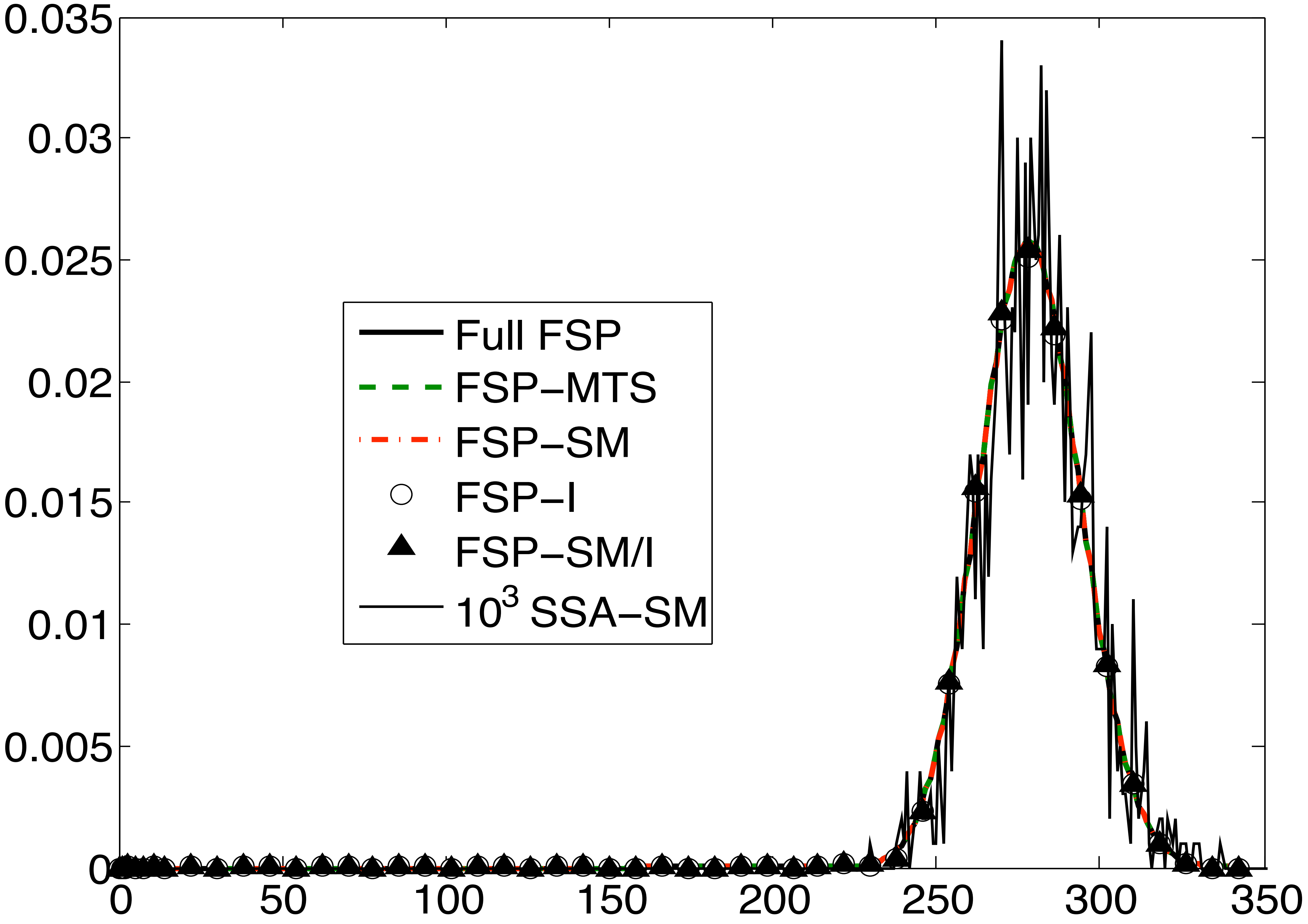
1. Full FSP
2. Slow manifold (FSP-SM)
3. Interpolated (FSP-I)
4. Hybrid (FSP-SM/I)
5. Multiple time interval (FSP-MTI)



70 sets of 195 or fewer ODEs.

Efficiency and accuracy of the reduced FSP methods

Brian Munsky



Efficiency and accuracy of the reduced FSP methods

For final time $t_f = 300s$				
Method	Matrix Size	J_{solve}	J_{total}	∞-norm Error
FSP	4459	750s	750s	$< 3.0 \times 10^{-5}$
FSP-MTS	195[1]	-	40.2s	$< 1.68 \times 10^{-4}$
FSP-SM	343	0.25s	0.94s	$\approx 5.1 \times 10^{-4}$
FSP-I	539	5.1s	6.1s	$\approx 7.7 \times 10^{-4}$
FSP-SM/I	49	0.04s	0.78s	$\approx 8.2 \times 10^{-4}$
10^4 SSA	Results would take more than 55 hours.			
10^3 SSA-SM	-	-	84.1s	≈ 0.0116
10^4 SSA-SM	-	-	925s	$\approx 3.4 \times 10^{-3}$
10^5 SSA-SM	-	-	9360s	$\approx 1.6 \times 10^{-3}$

The Reduced FSP approaches are much faster and more accurate than alternative approaches!

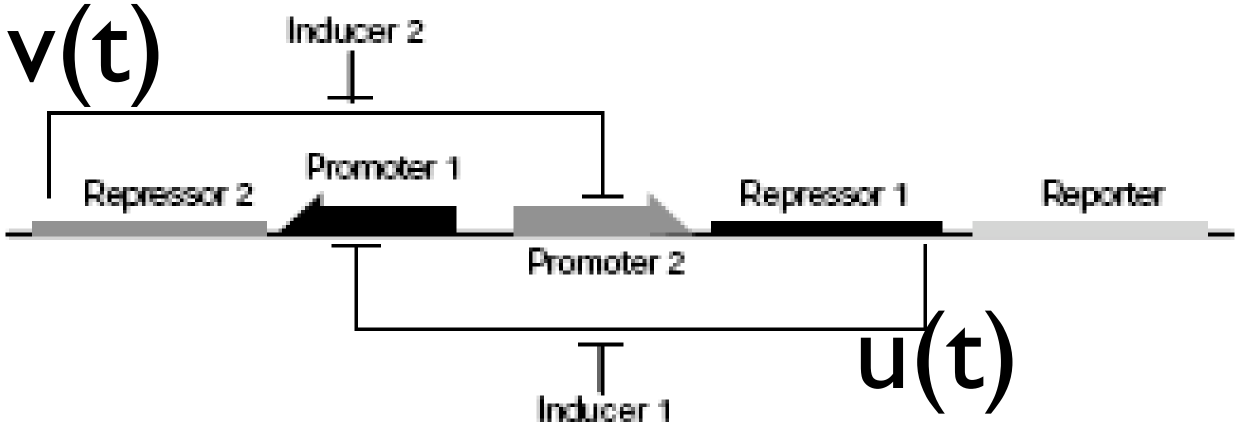
Outline

- ☑ Introduction
- ☑ Monte Carlo Solution Schemes
- ☑ Finite State Projection (FSP)
- ☑ Reductions to the FSP
  - Example: Genetic Toggle Switch
    1. SSA and FSP analysis
    2. Switch and trajectory analysis
    3. Sensitivity and Model Identification.

Genetic Toggle Model:

Gardner, *et al.*, *Nature* 403, 339-342 (2000)

Brian Munsky



Two repressors, *u* and *v*.

v inhibits the production of *u*.

u inhibits the production of *v*.

Both *u* and *v* degrade exponentially.

$$a_1(u,v) = \frac{\alpha_1}{1+v^\beta} \quad \nu_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$a_3(u,v) = \frac{\alpha_2}{1+u^\gamma} \quad \nu_3 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$a_2(u,v) = u \quad \nu_2 = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$$

$$a_4(u,v) = v \quad \nu_4 = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$$

$$\alpha_1 = 50 \quad \beta = 2.5$$

$$\alpha_2 = 16 \quad \gamma = 1$$

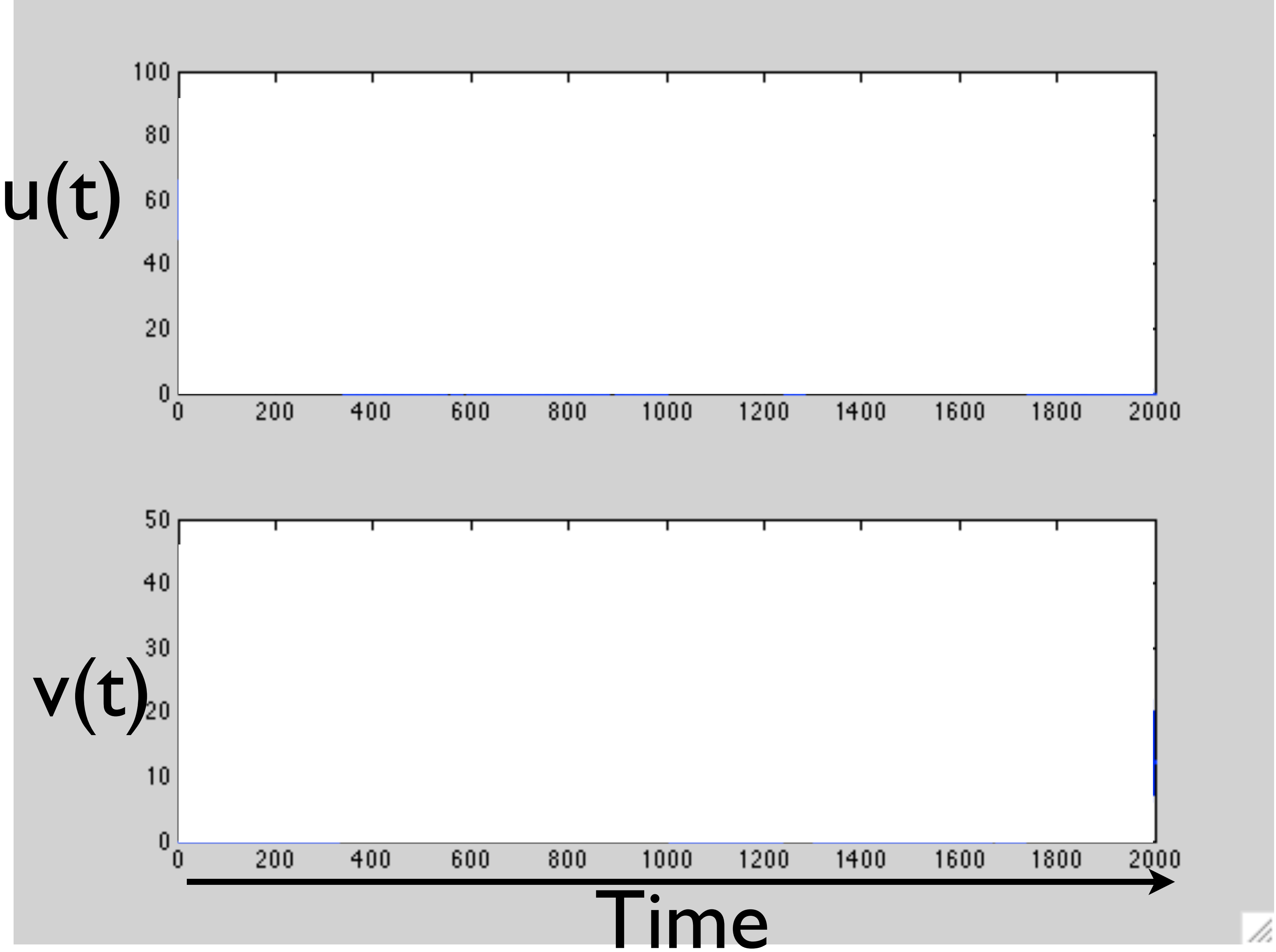
A Sample Trajectory

Brian Munsky

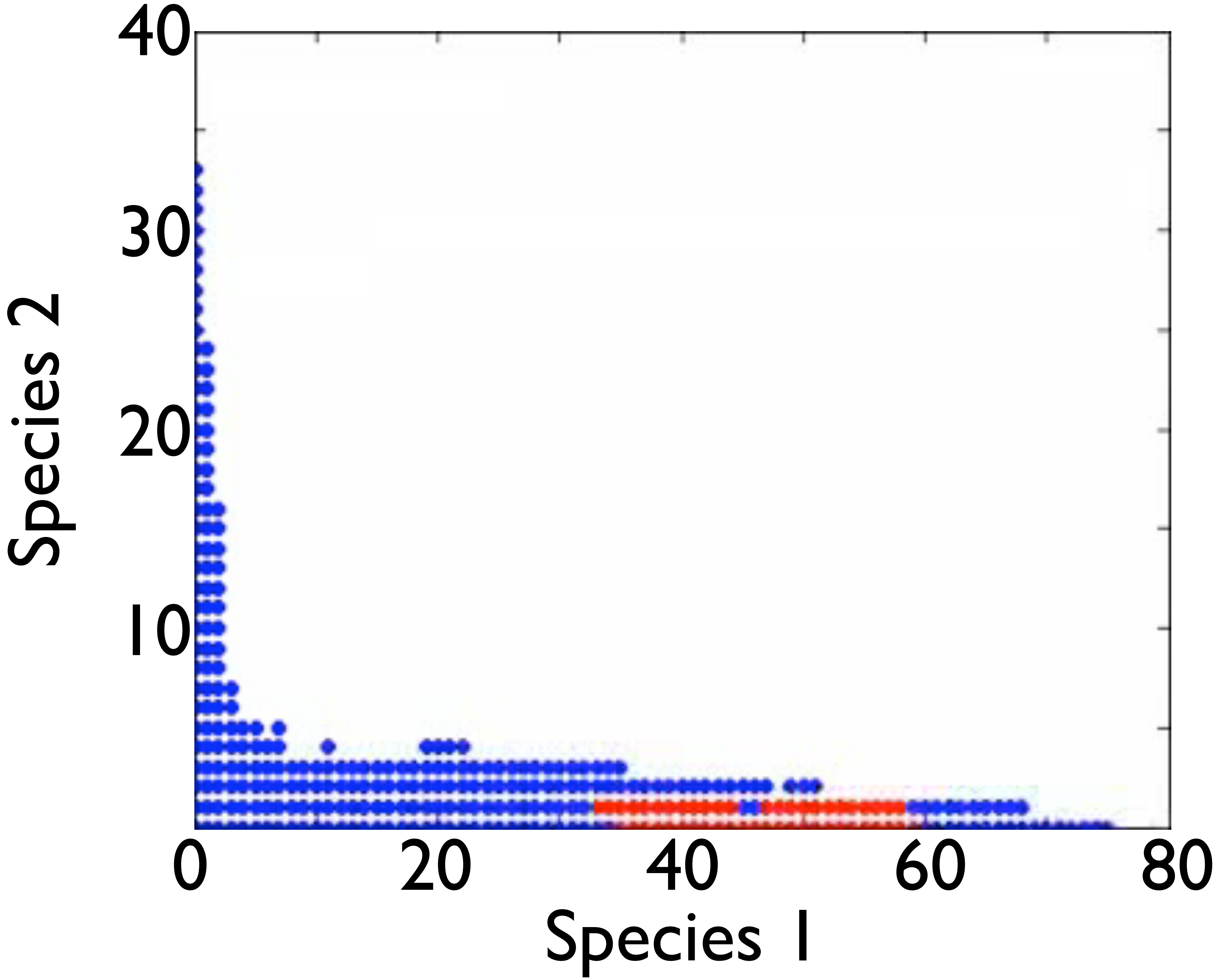
We begin with an initial condition:

$$\left[\begin{array}{c} u(t) \\ v(t) \end{array}\right] = \left[\begin{array}{c} 60 \\ 0 \end{array}\right]$$

and consider a sample trajectory.

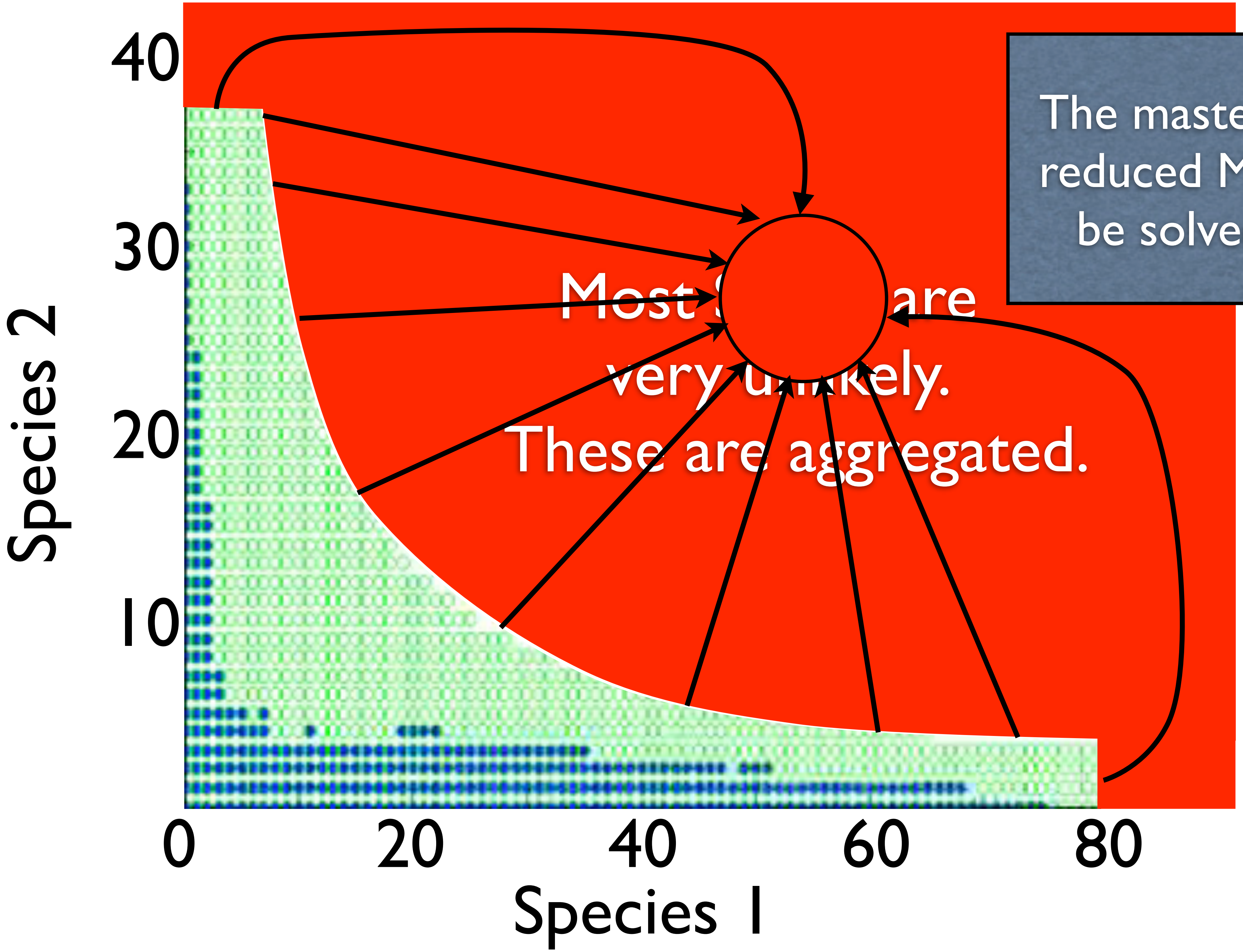


Choosing the Finite State Projection



Choosing the Finite State Projection

Brian Munsky

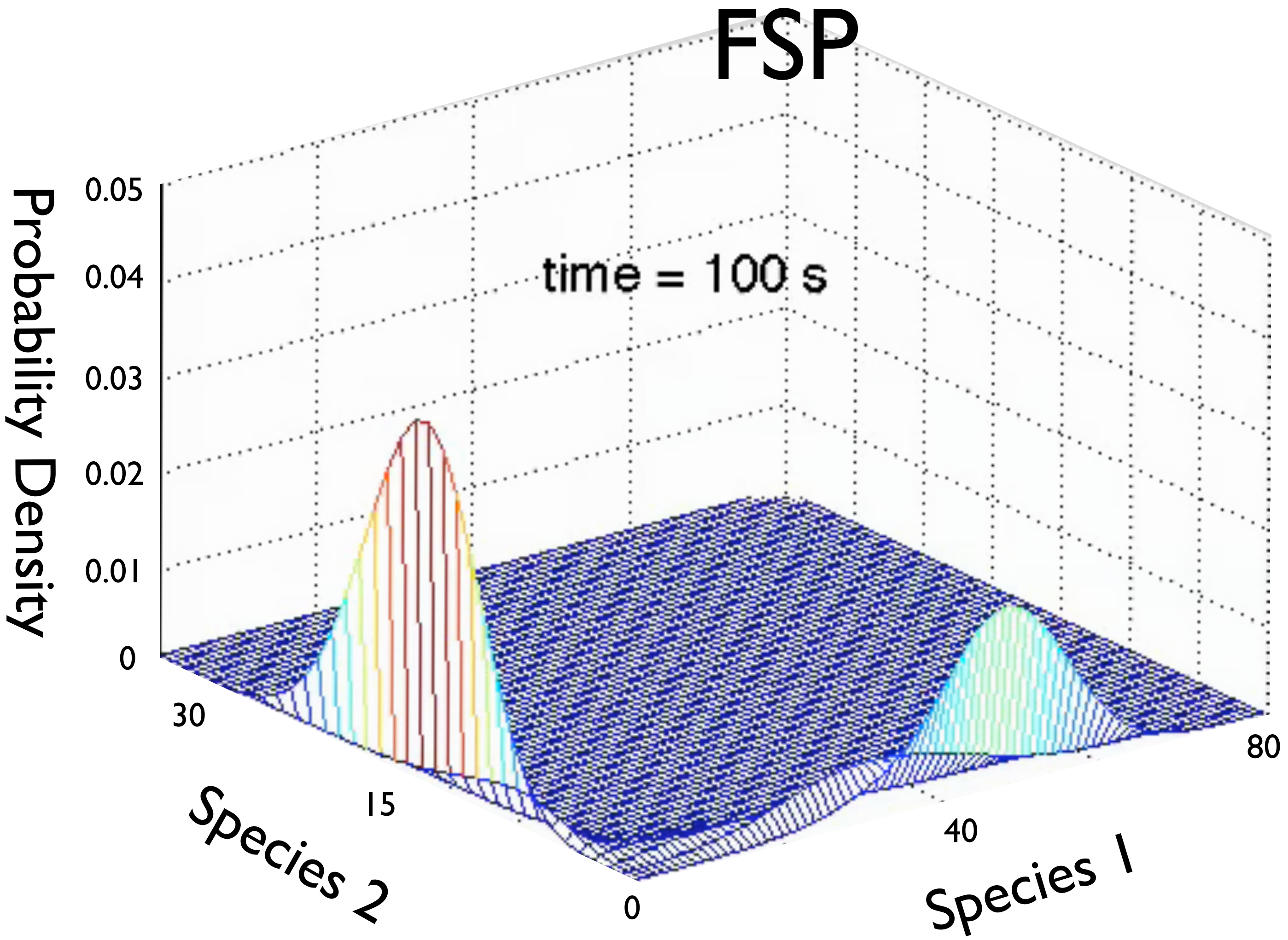


Most S[illegible] are very u[illegible]kely.

These are aggregated.

The master equation for this reduced Markov process can be solved very efficiently.

The Toggle Switch Distribution



FSP

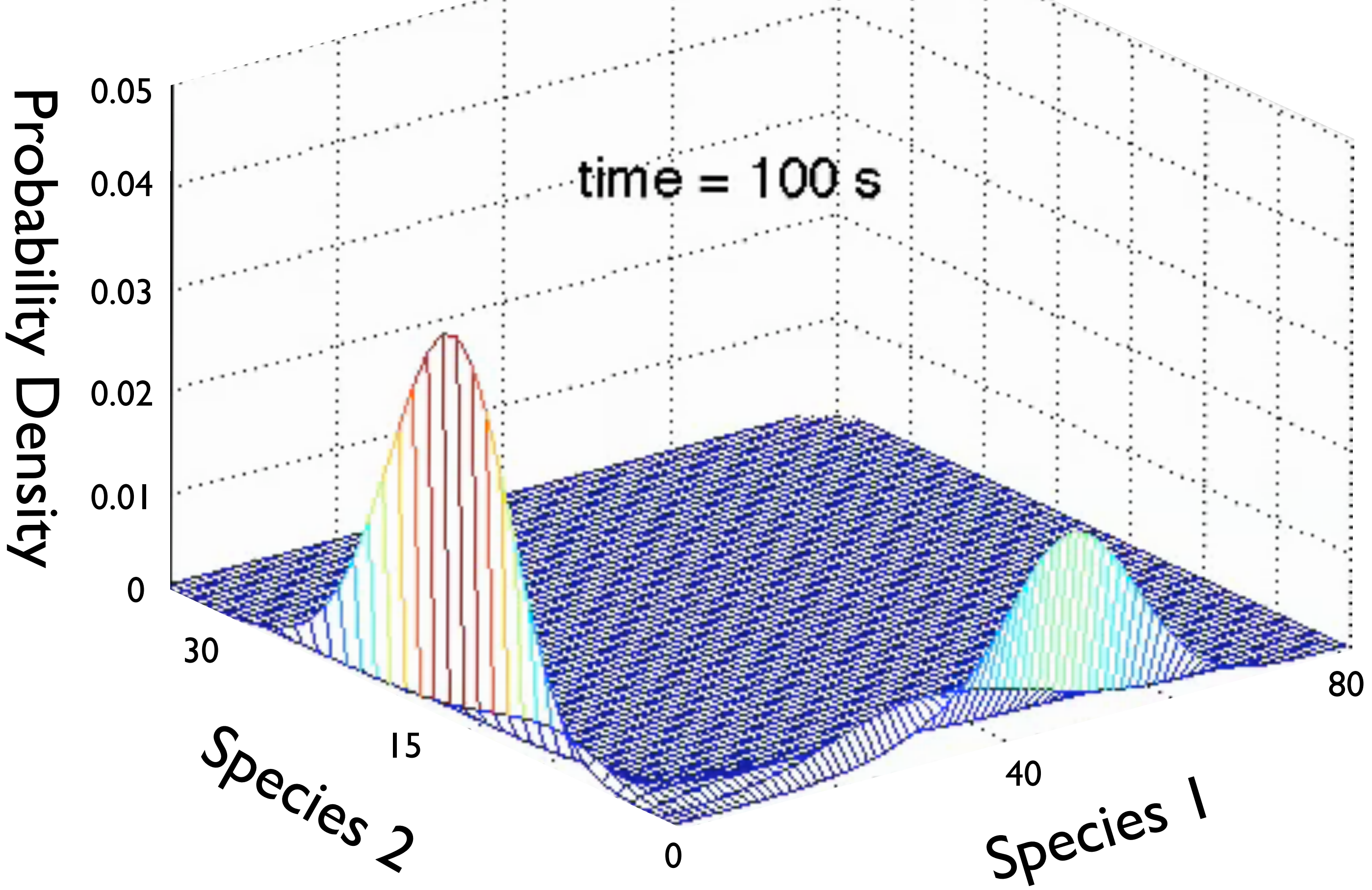
Method	# Simulations	Time (s)	$\|\|\text{Error}\|\|_1$
FSP	– [a]	5	$\leq 12 \times 10^{-3}$

Guaranteed

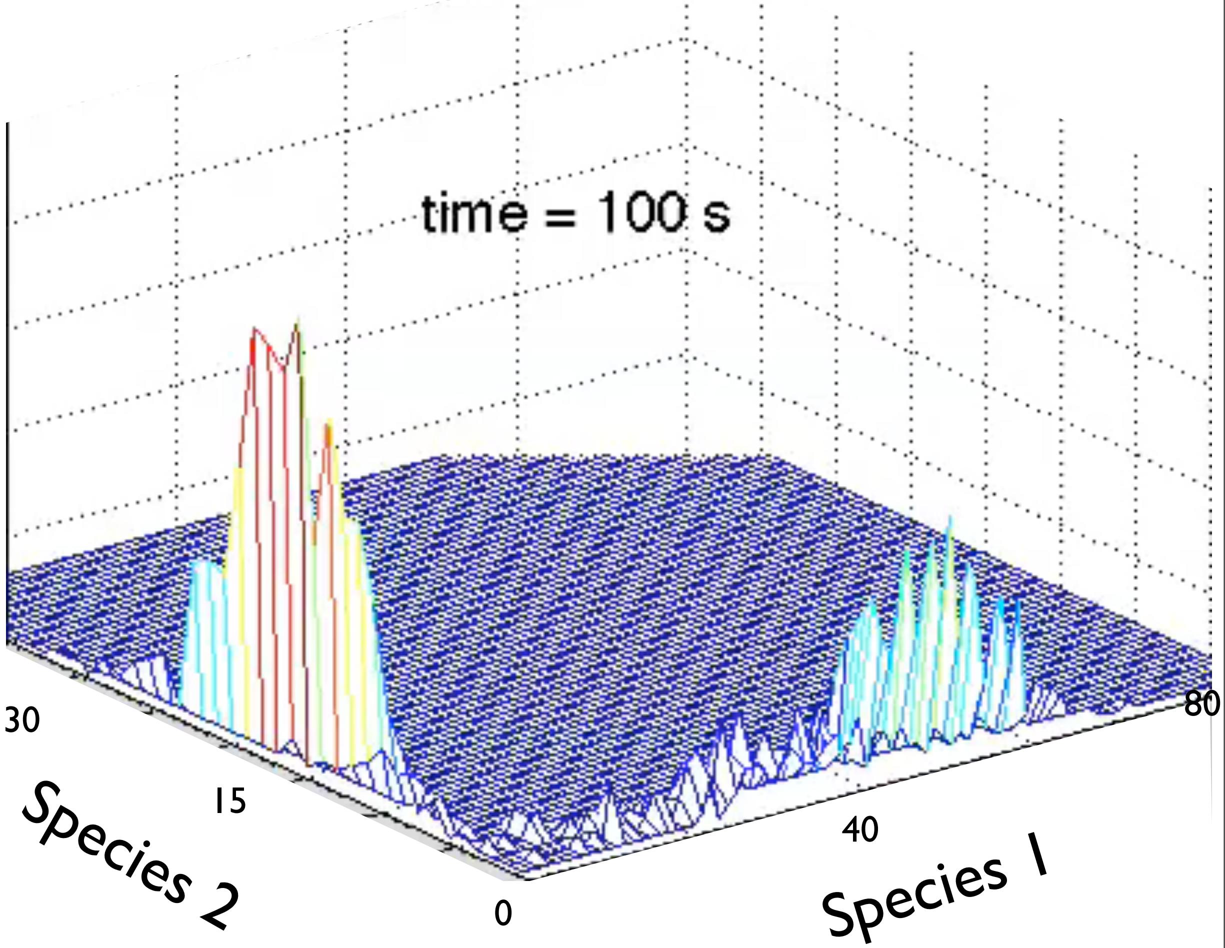
[a] The FSP algorithm is run only once.

The Toggle Switch Distribution

FSP



SSA (1000 runs)



Method	# Simulations	Time (s)	$\|\|\text{Error}\|\|_1$
FSP	$-$[a]	5	$\leq 12 \times 10^{-3}$
SSA	10^3	108	≈ 0.33

Guaranteed

No Guarantees

[a] The FSP algorithm is run only once.

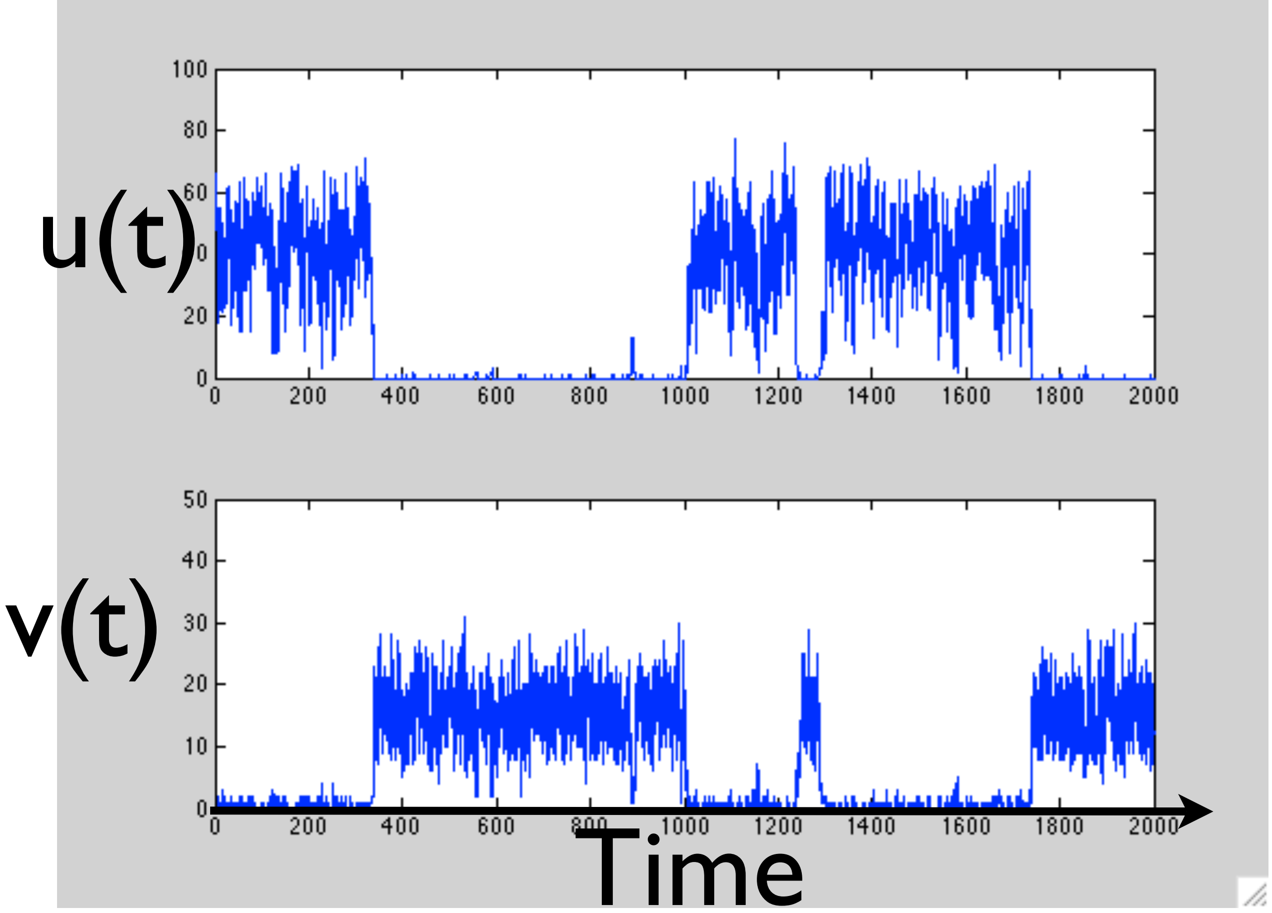
Switch rates of the gene toggle model.

Switch Analysis

Define the switch to be OFF when *v*(*t*) > 5 and *u*(*t*) < 16 and ON when *v*(*t*) < 5 and *u*(*t*) > 16.

We begin with an initial condition, $\left[\begin{array}{c} u(0) \\ v(0) \end{array}\right] = \left[\begin{array}{c} 0 \\ 0 \end{array}\right]$, and ask a few questions:

1. What portion will turn OFF first (before they turn ON?
2. How long until 99% of trajectories will make this first decision?
3. How long until 99% of trajectories will turn ON?
4. How long until 50% of trajectories will turn OFF first then ON?



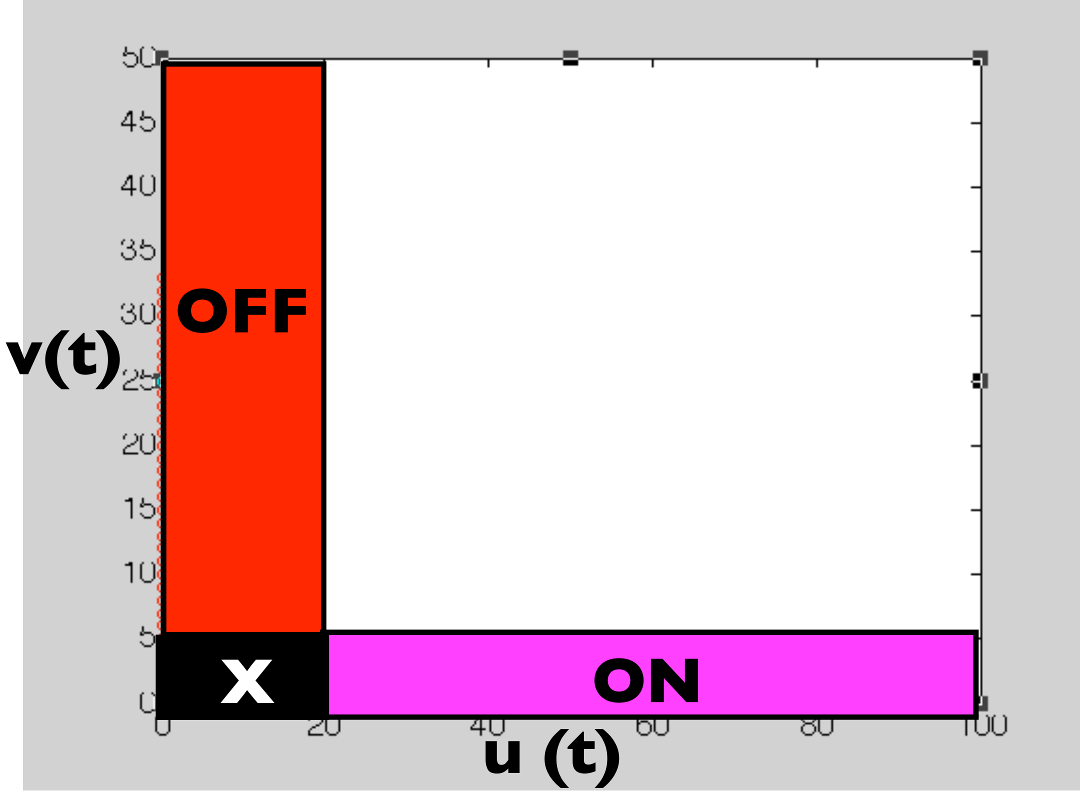
(1) Direction of First Switch

We define some configuration subsets:

OFF - absorbing region corresponding to trajectories that have entered the OFF region.

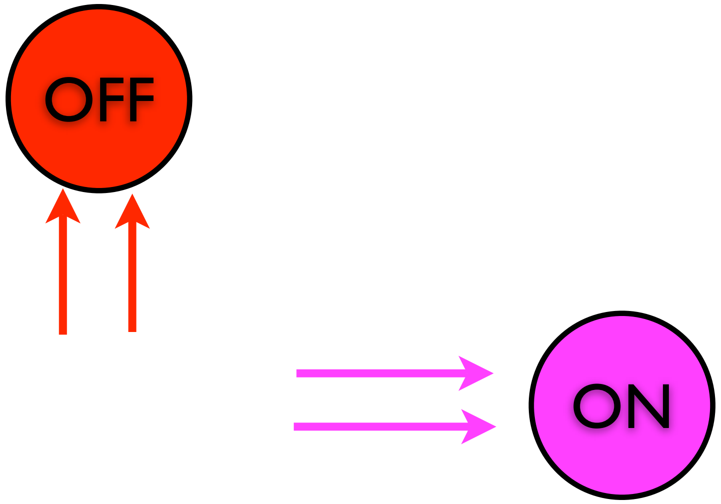
ON - absorbing region corresponding to trajectories that have entered the ON region.

X - every other reachable state.



Aggregate OFF and ON.

Keep reactions originating in X, but remove the rest.

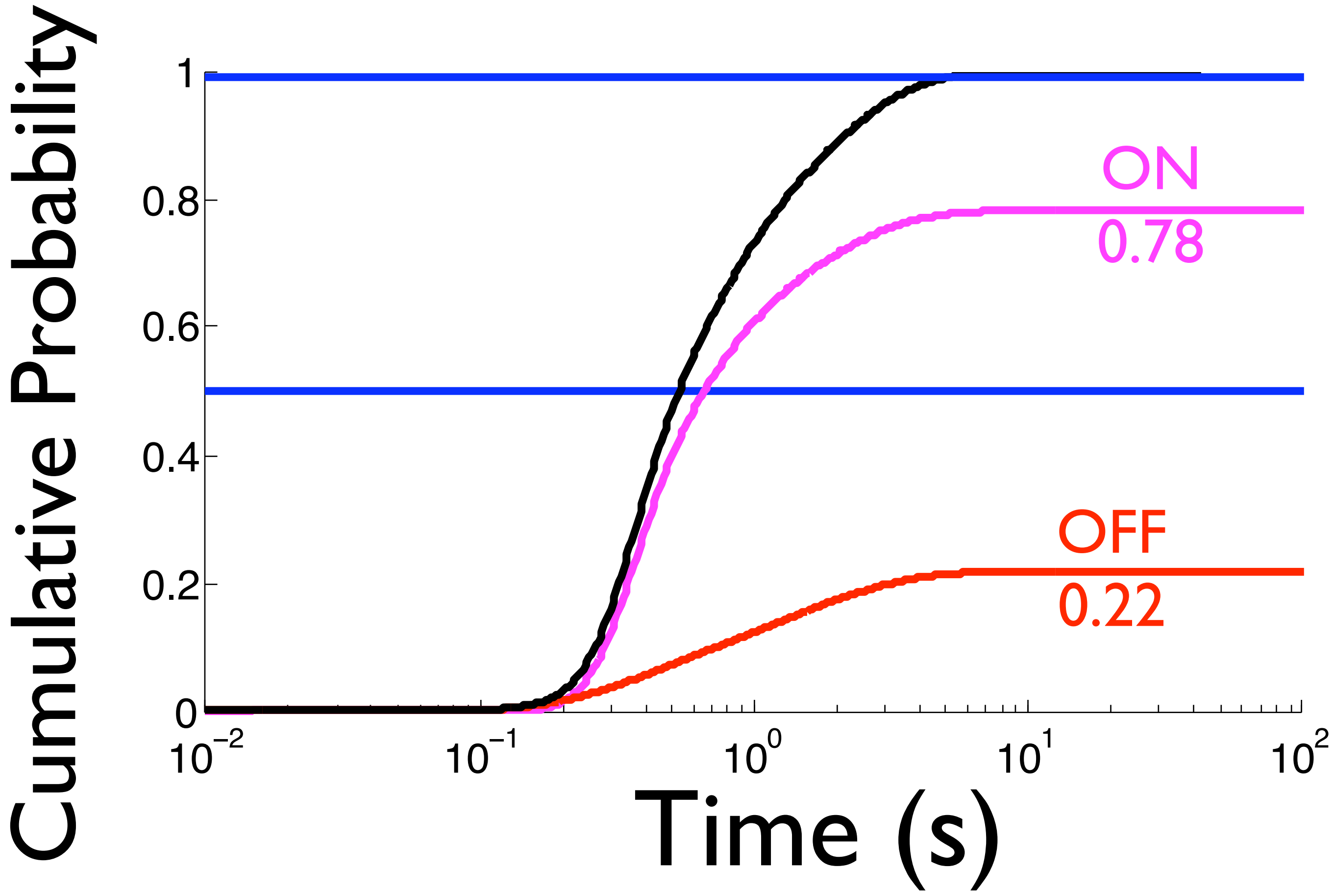


Solve for OFF(t) and ON(t)

(1) Direction of First Switch

Probability of Turning
ON first: 0.78

Probability of Turning
OFF first: 0.22



(2) 50% and 99% Time of First Switch

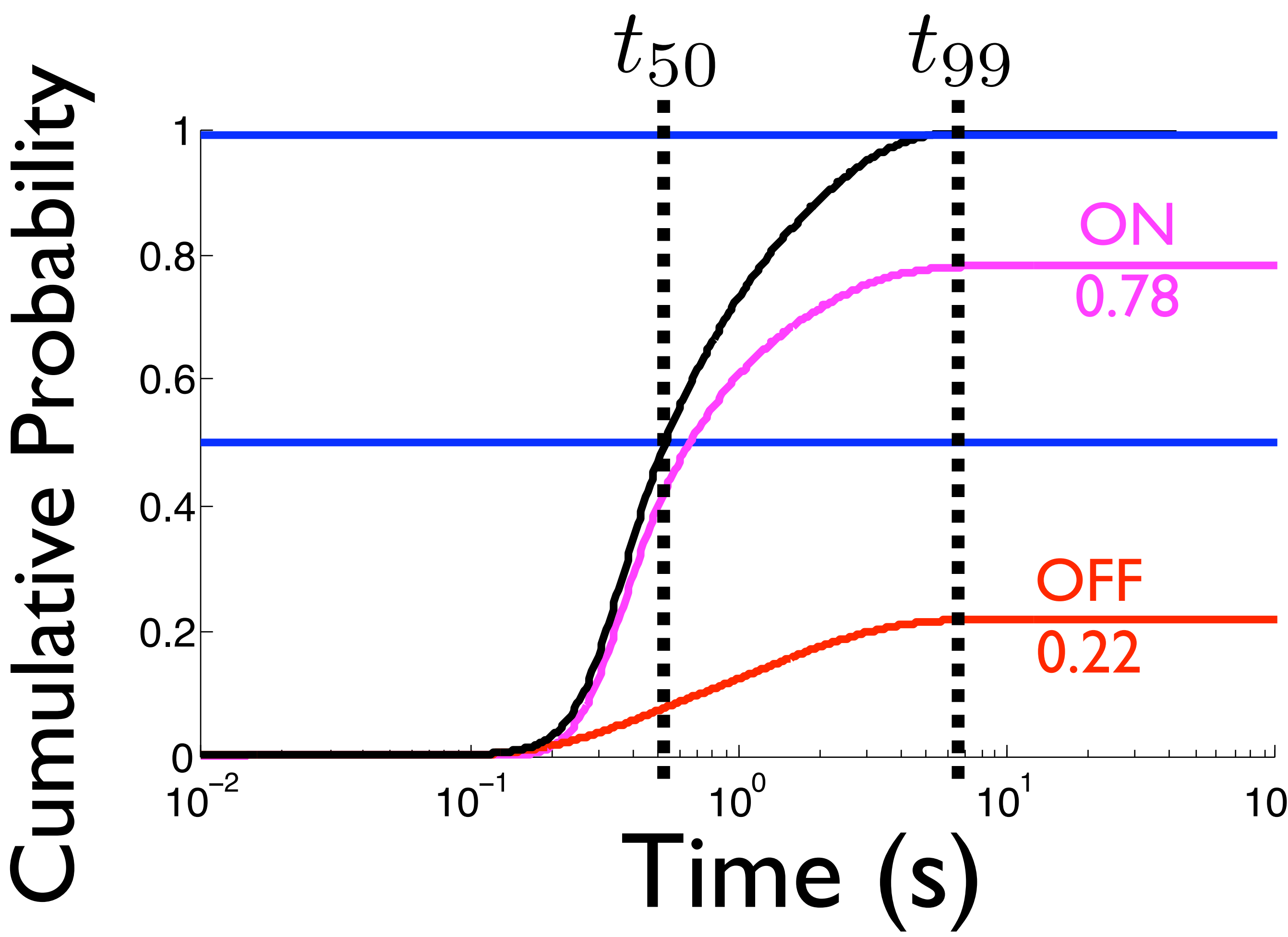
Brian Munsky

Probability of Turning ON first: 0.78

Probability of Turning OFF first: 0.22

$$t_{50} = 0.5305s$$

$$t_{99} = 5.0595s$$



(3) 99% Time of first OFF switch

Brian Munsky

We define some configuration subsets:

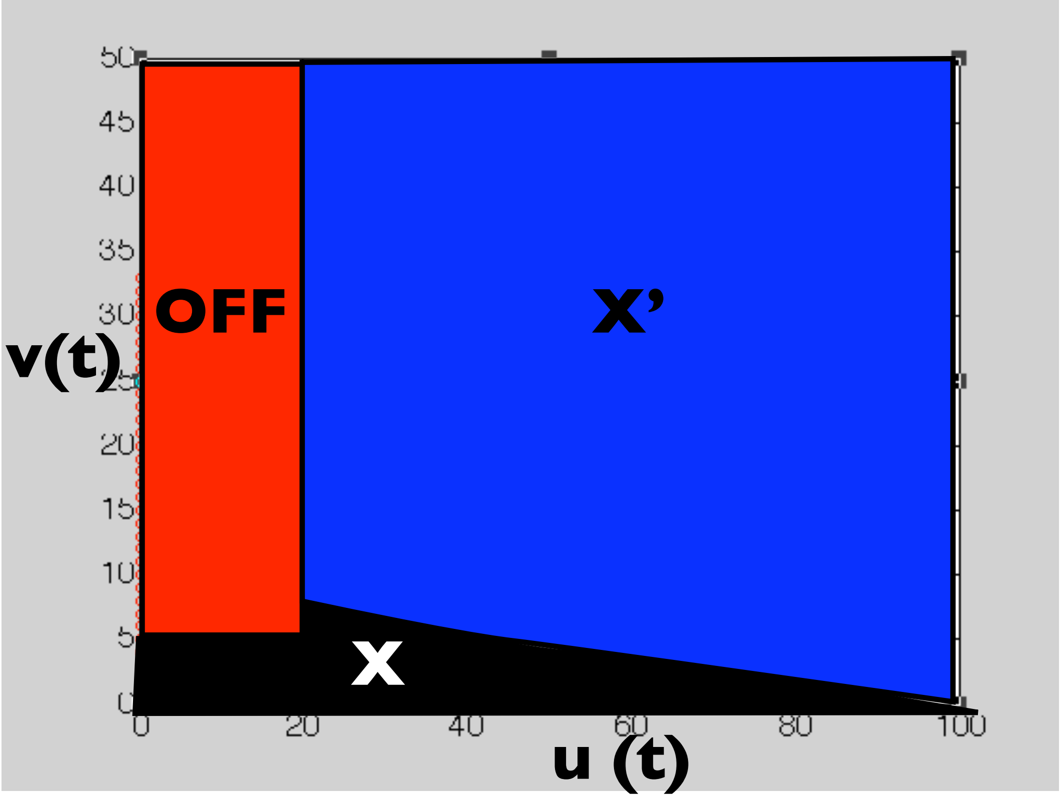
OFF - absorbing region corresponding to trajectories that have entered the OFF region.

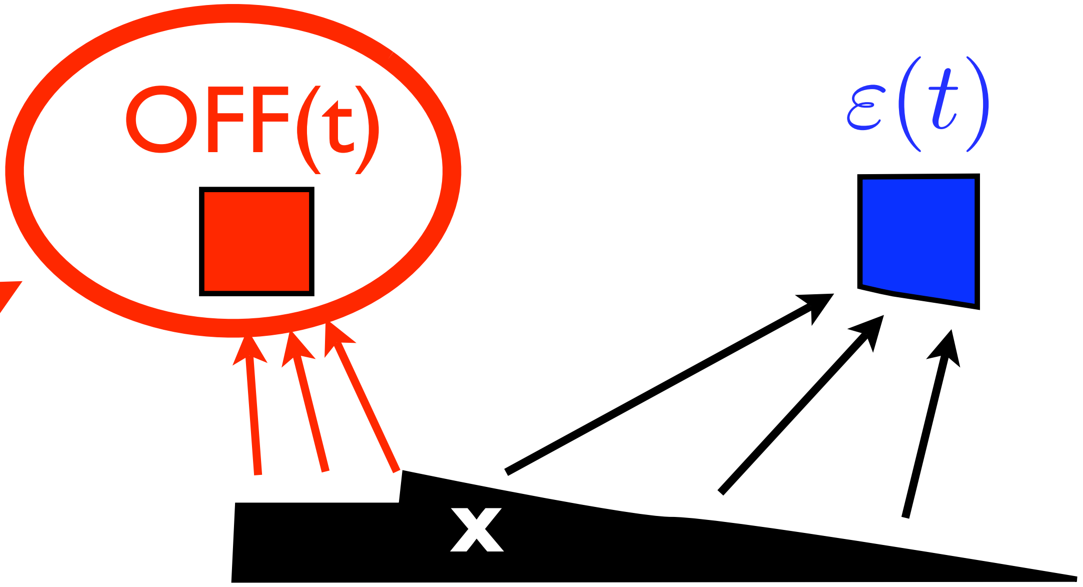
X' - unlikely states.

X - everything else.

Aggregate OFF and X'.

Keep reactions originating in X, but remove the rest.





Solve for OFF(t)

(3) 99% Time of first OFF switch

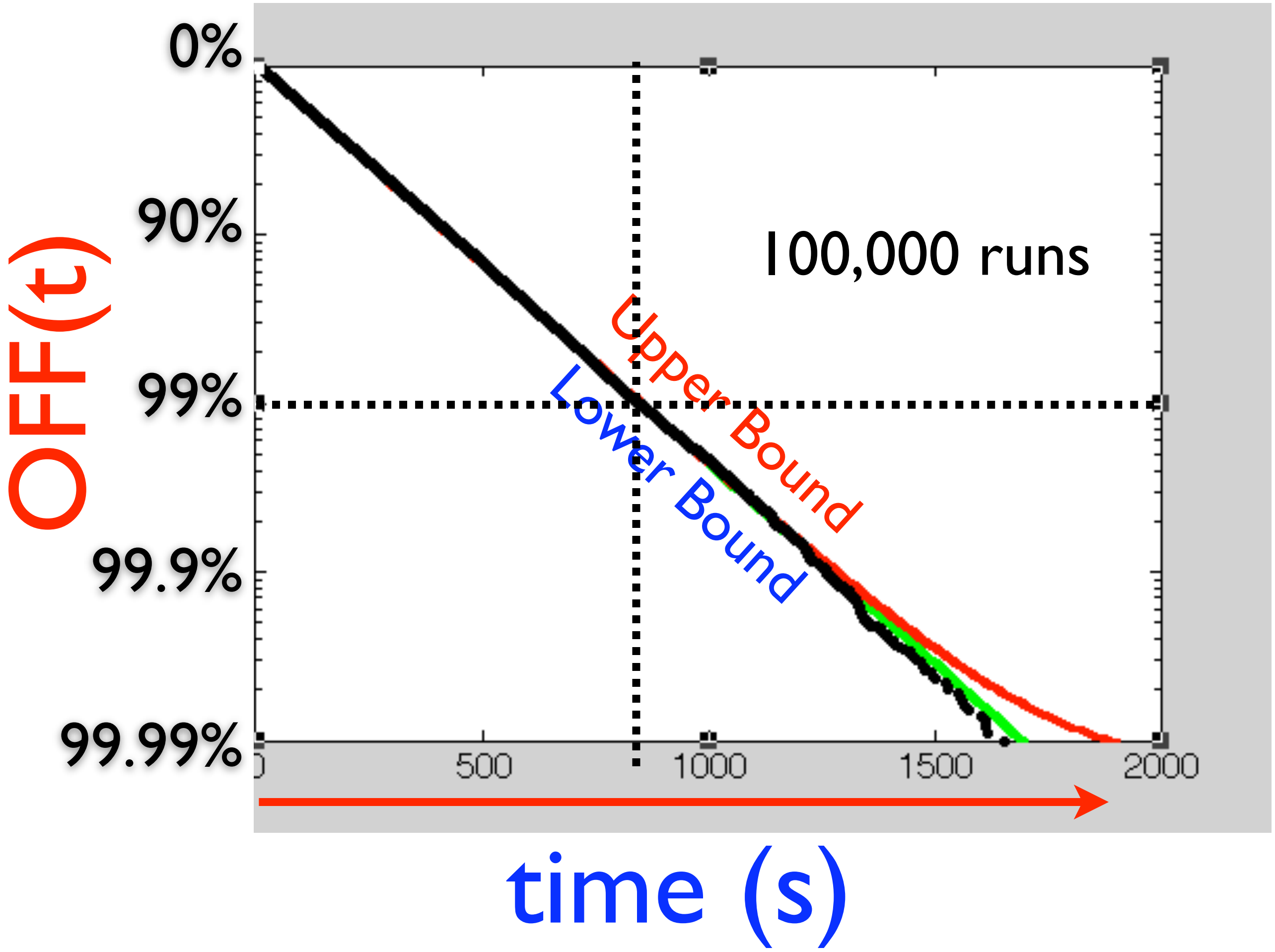
Brian Munsky

Provides guaranteed bounds on the probability of switching.

Monte Carlo simulations (SSA) require many many runs to achieve comparable precision.

Provide no accuracy guarantees.

Probability of turning OFF vs. time



The FSP approach also provides estimates of every other initial probability distribution supported on $\mathbf{X}_J$.
Monte Carlo methods only consider a single initial distribution.

FSP vs. Monte Carlo Algorithms

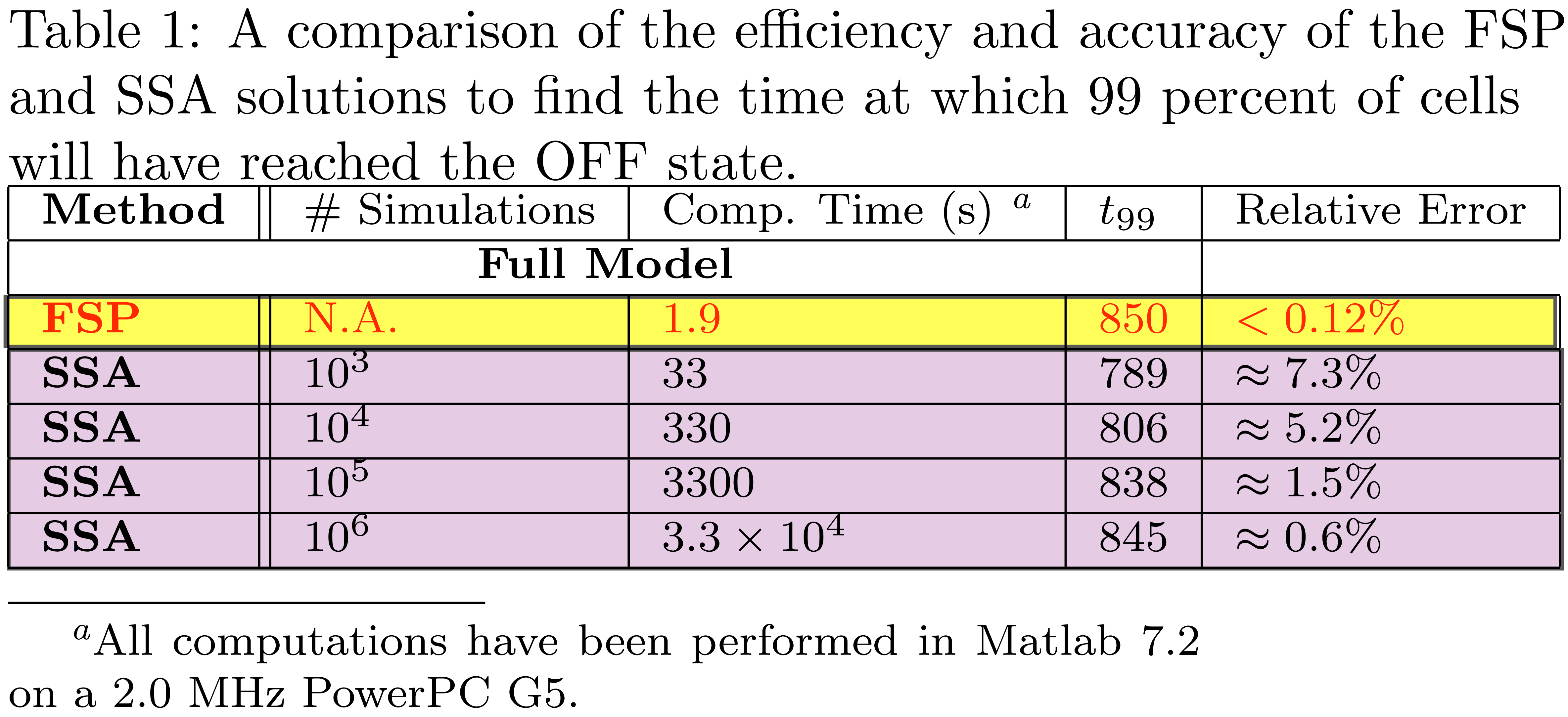


Table 1: A comparison of the efficiency and accuracy of the FSP and SSA solutions to find the time at which 99 percent of cells will have reached the OFF state.

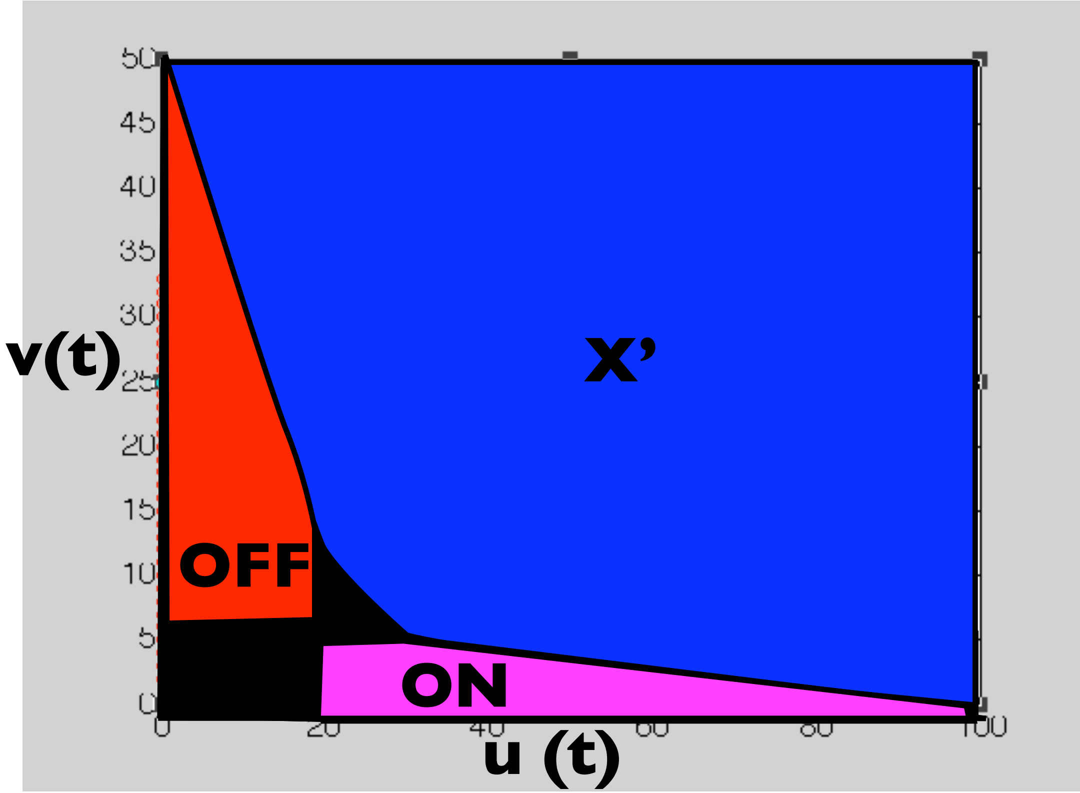
Method	# Simulations	Comp. Time (s) [a]	t_{99}	Relative Error
		Full Model		
FSP	N.A.	1.9	850	$< 0.12\%$
SSA	10^3	33	789	$\approx 7.3\%$
SSA	10^4	330	806	$\approx 5.2\%$
SSA	10^5	3300	838	$\approx 1.5\%$
SSA	10^6	3.3×10^4	845	$\approx 0.6\%$

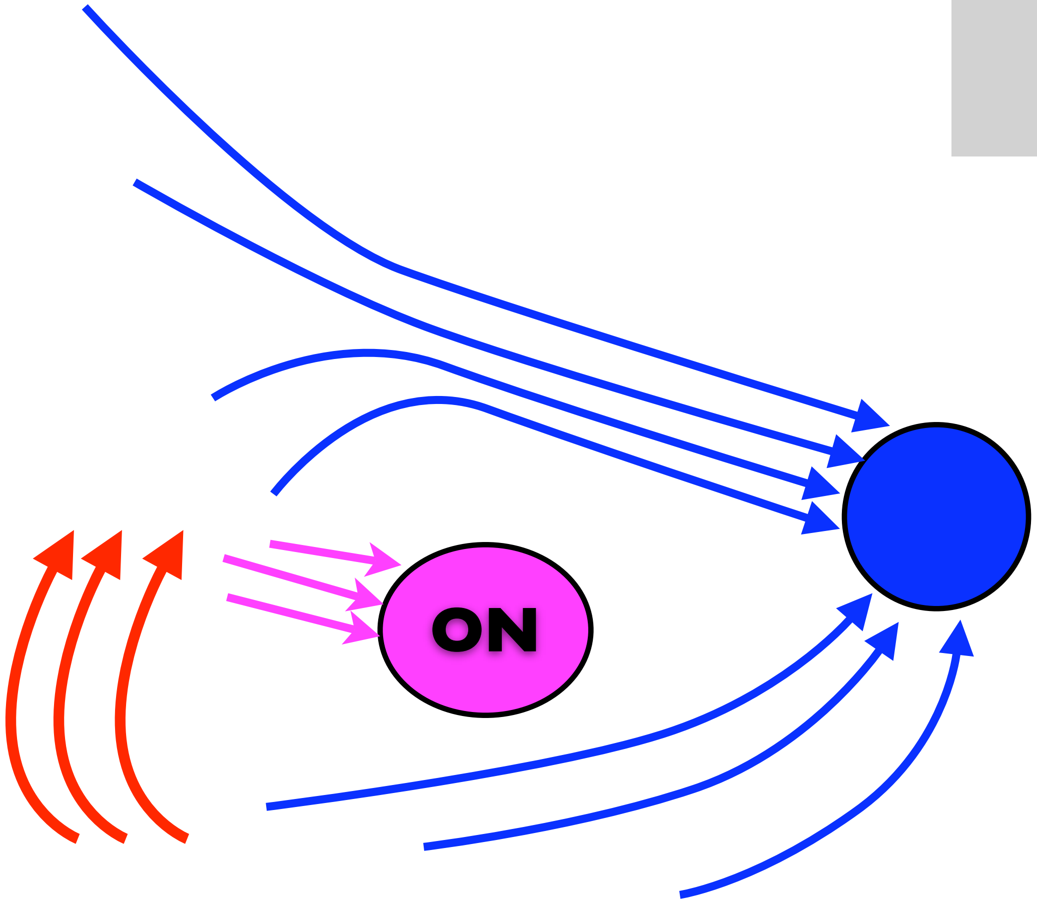
[a] All computations have been performed in Matlab 7.2 on a 2.0 MHz PowerPC G5.

(4) Median Time of first OFF then ON trajectory

Stages:

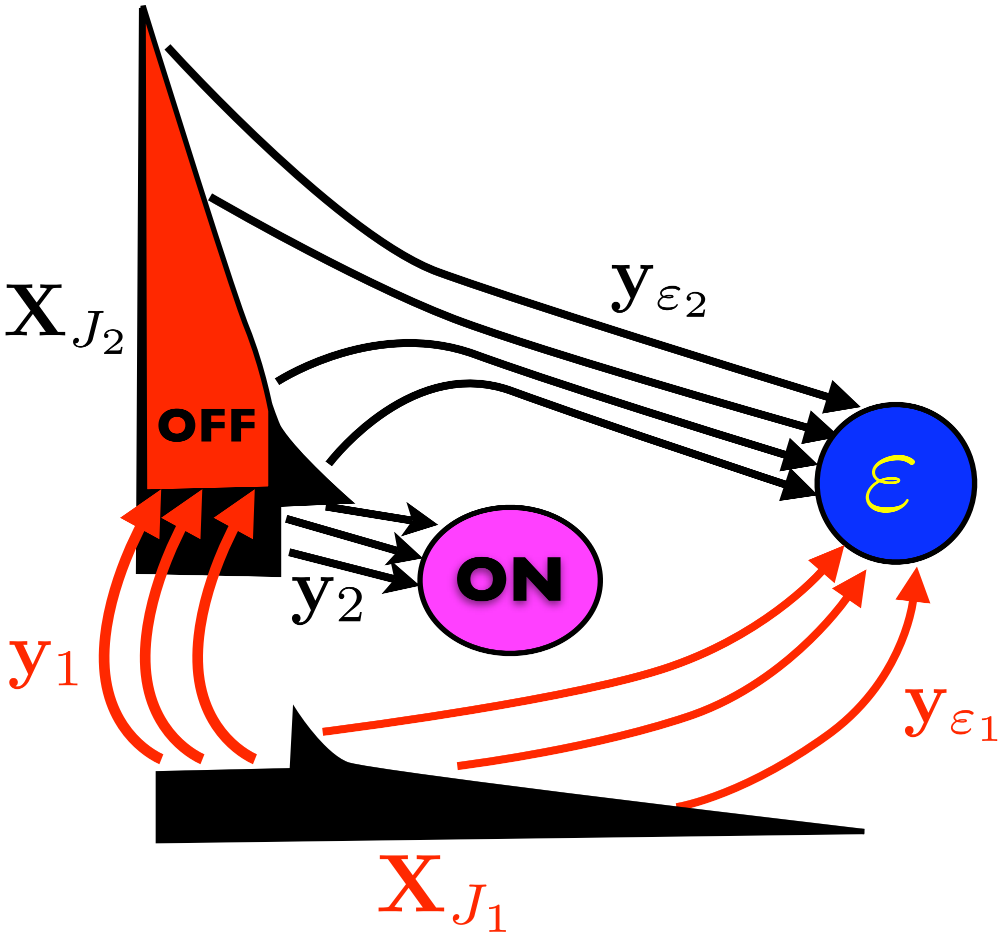
1. Remain in set of all *not OFF* states until switch to OFF.
2. Remain in set of all *not ON* states until switch to ON.





(4) Median Time of first OFF then ON trajectory

Brian Munsky



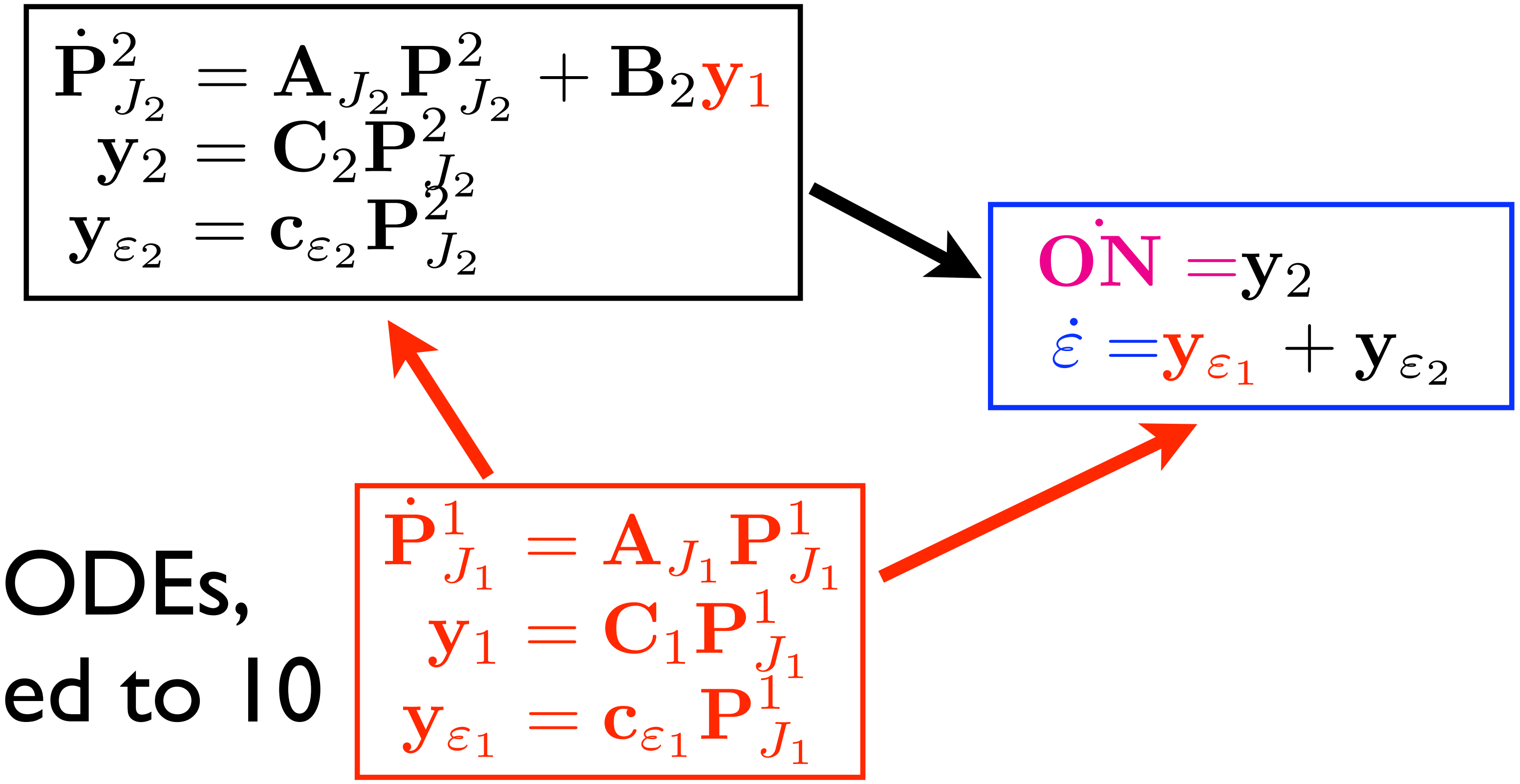
$$\begin{bmatrix}\dot{\mathbf{P}}^1_{J_1}\\ \dot{\mathbf{P}}^2_{J_2}\\ \dot{ON}\\ \varepsilon\end{bmatrix} = \begin{bmatrix}\mathbf{A}_{J_1} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{B}_2\mathbf{C}_1 & \mathbf{A}_{J_2} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{C}_2 & \mathbf{0} & \mathbf{0}\\ \mathbf{c}_{\varepsilon_1} & \mathbf{c}_{\varepsilon_2} & \mathbf{0} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{P}^1_{J_1}\\ \mathbf{P}^2_{J_2}\\ ON\\ \varepsilon\end{bmatrix}$$

$$\begin{aligned}\dot{\mathbf{P}}^2_{J_2} &= \mathbf{A}_{J_2}\mathbf{P}^2_{J_2} + \mathbf{B}_2\mathbf{y}_1\\ \mathbf{y}_2 &= \mathbf{C}_2\mathbf{P}^2_{J_2}\\ \mathbf{y}_{\varepsilon_2} &= \mathbf{c}_{\varepsilon_2}\mathbf{P}^2_{J_2}\end{aligned}$$

$$\begin{aligned}\dot{ON} &= \mathbf{y}_2\\ \dot{\varepsilon} &= \mathbf{y}_{\varepsilon_1} + \mathbf{y}_{\varepsilon_2}\end{aligned}$$

$$\begin{aligned}\dot{\mathbf{P}}^1_{J_1} &= \mathbf{A}_{J_1}\mathbf{P}^1_{J_1}\\ \mathbf{y}_1 &= \mathbf{C}_1\mathbf{P}^1_{J_1}\\ \mathbf{y}_{\varepsilon_1} &= \mathbf{c}_{\varepsilon_1}\mathbf{P}^1_{J_1}\end{aligned}$$

Hankel Norm Reduction (Balanced Truncation)



780 ODEs,
Reduced to 10

$$\dot{\mathbf{P}}^2_{J_2} = \mathbf{A}_{J_2}\mathbf{P}^2_{J_2} + \mathbf{B}_2\mathbf{y}_1$$
$$\mathbf{y}_2 = \mathbf{C}_2\mathbf{P}^2_{J_2}$$
$$\mathbf{y}_{\varepsilon_2} = \mathbf{c}_{\varepsilon_2}\mathbf{P}^2_{J_2}$$

$$\dot{\mathrm{ON}} = \mathbf{y}_2$$
$$\dot{\varepsilon} = \mathbf{y}_{\varepsilon_1} + \mathbf{y}_{\varepsilon_2}$$

2 ODEs

714 ODEs,
Reduced to 10

$$\dot{\mathbf{P}}^1_{J_1} = \mathbf{A}_{J_1}\mathbf{P}^1_{J_1}$$
$$\mathbf{y}_1 = \mathbf{C}_1\mathbf{P}^1_{J_1}$$
$$\mathbf{y}_{\varepsilon_1} = \mathbf{c}_{\varepsilon_1}\mathbf{P}^1_{J_1}$$

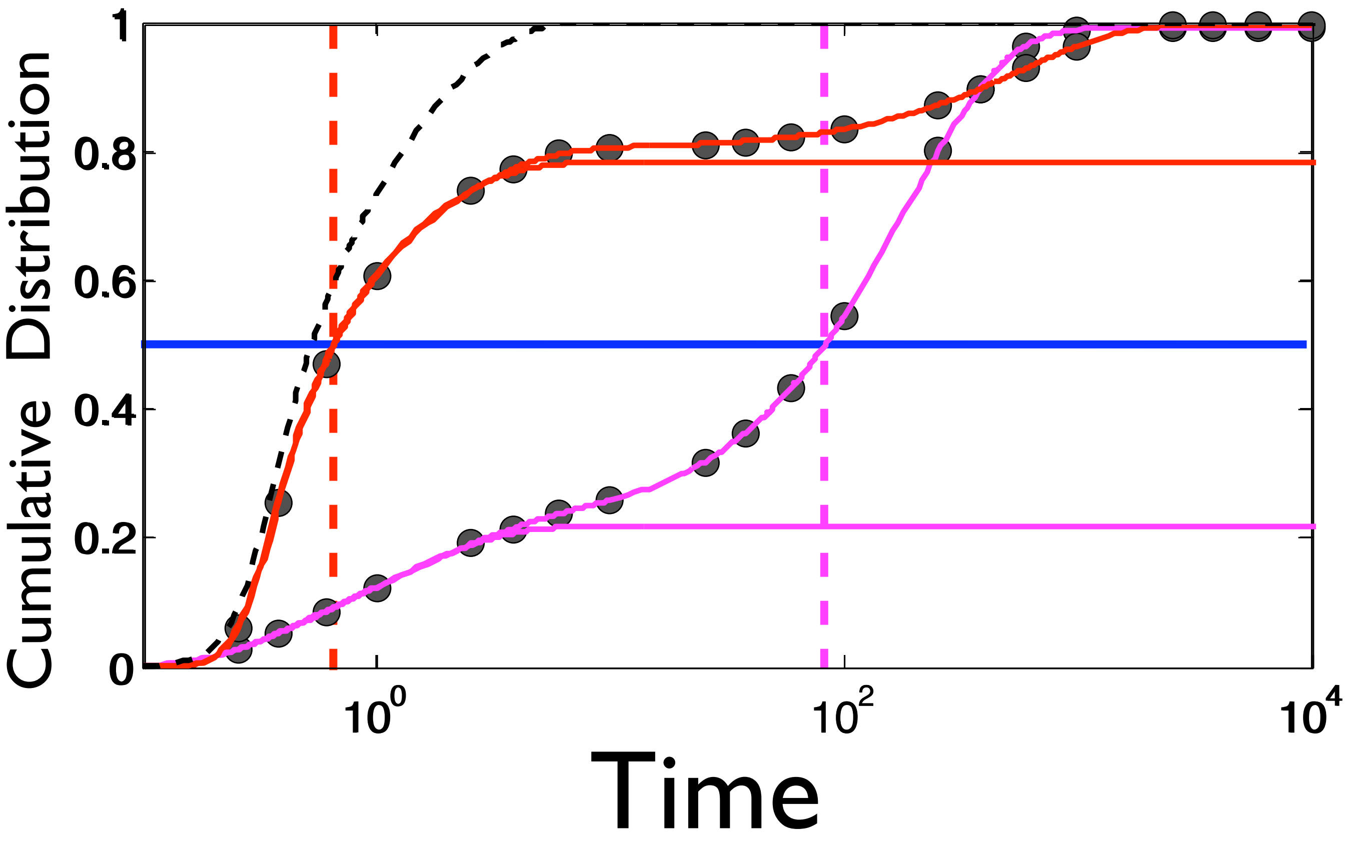
Total: 1496 ODEs

Reduced to 22

Further reductions are possible.

Median Times

1. First Switch: 0.5305s
2. First ON: 0.6565s
3. First OFF: 81.952s



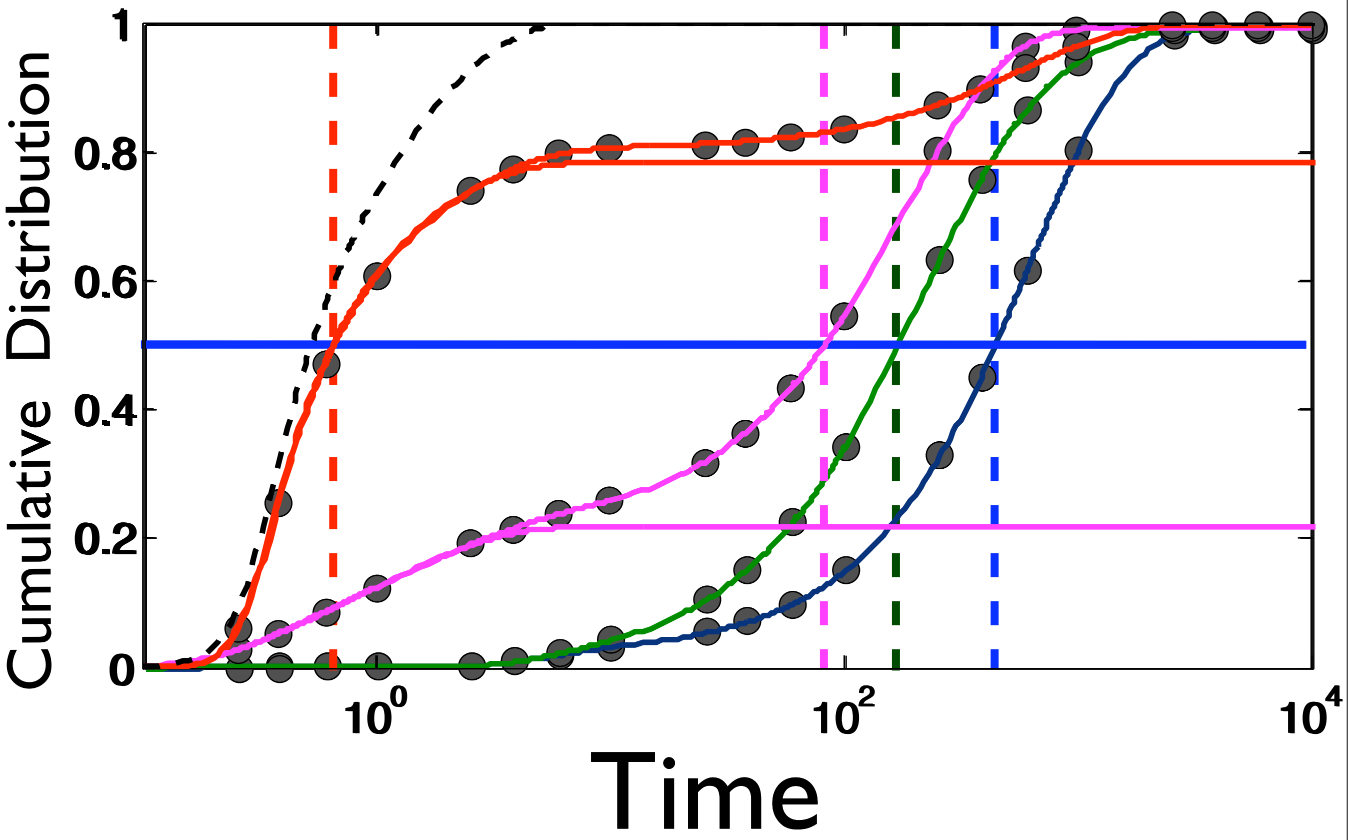
- Observations:
  - The first decision is ON more often than OFF.
  - The OFF region is more stable than the ON region.
  - Reduced models capture switching very accurately.

Median Times

Brian Munsky

1. First Switch: 0.5305s
2. First ON: 0.6565s
3. First OFF: 81.952s
4. First ON then OFF: 167.530s
5. First OFF then ON: 434.969s

- Observations:
  - The first decision is ON more often than OFF.
  - The OFF region is more stable than the ON region.
  - Reduced models capture switching very accurately.



Errors are guaranteed to be less than line thickness!

Both the 1496-order FSP and the 22-order FSP-RED approaches yield very accurate results.

After the reduction the 22-order FSP-RED approach is far more efficient.

At present, however, the reduction is quite computationally expensive.

Single Stage Trajectories

First Switch to OFF

Method	J_{red}	J_{solve}	J_{total}[a]	t_{50}	% Error
FSP	-	31.0s	31.0s	$81.952s$	$< 2 \times 10^{-5}$
FSP-RED	111.8	1.85s	113.7s	$81.952s$	$< 4 \times 10^{-5}$
10^4 SSA	-	2068s	2068s	78.375s	≈ 4.3

First Switch to ON

	J_{red}	J_{solve}	J_{total}	t_{50}	% Error
FSP	-	25.7s	25.7s	$0.65655s$	$< 1 \times 10^{-7}$
FSP-RED	133.5s	1.85s	135.3s	$0.65656s$	$< 8 \times 10^{-4}$
10^4 SSA	-	404.4s	404.4	0.65802s	≈ 0.22

Two Stage Trajectories

First Completion of OFF then ON trajectory

	J_{red}	J_{solve}	J_{total}	t_{50}	% Error
FSP	-	46.9s	46.9s	$434.969s$	$< 3.5 \times 10^{-5}$
FSP-RED	222.0s	1.95s	224.0s	$434.968s$	$< 4.5 \times 10^{-3}$
10^4 SSA	-	3728s	3728s	441.394	≈ 1.5

First Completion of ON then OFF trajectory

	J_{red}	J_{solve}	J_{total}	t_{50}	% Error
FSP	-	51.0s	51.0s	$167.530s$	$< 6 \times 10^{-7}$
FSP-RED	241.4s	1.98s	243.4s	167.939	≈ 0.24
10^4 SSA	-	3073s	3073	166.860	≈ 0.40

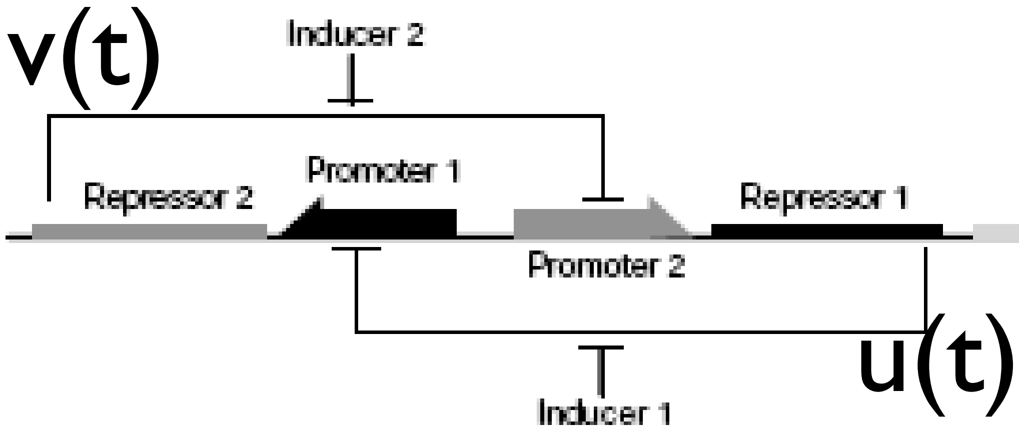
[a] All simulations have been performed in MATLAB version R2007a on a MacBook Pro with a 2.16 GHz Intel Core Duo processor and 2 GB of memory. All ODEs have been solved with MATLAB's stiff ODE solver *ode15s* with relative tolerance 10^{-8} and absolute tolerance of 10^{-20}.

Sensitivity Analysis and Model Identification for the gene toggle switch.

Genetic Toggle Model:

Gardner, *et al.*, *Nature* 403, 339-342 (2000)

Brian Munsky



Two repressors, *u* and *v*.

v inhibits the production of *u*.

u inhibits the production of *v*.

Both *u* and *v* degrade exponentially.

$$a_1(u,v) = \frac{\alpha_1}{1+v^\beta} \qquad \nu_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$a_3(u,v) = \frac{\alpha_2}{1+u^\gamma} \qquad \nu_3 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$a_2(u,v) = u \qquad \nu_2 = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$$

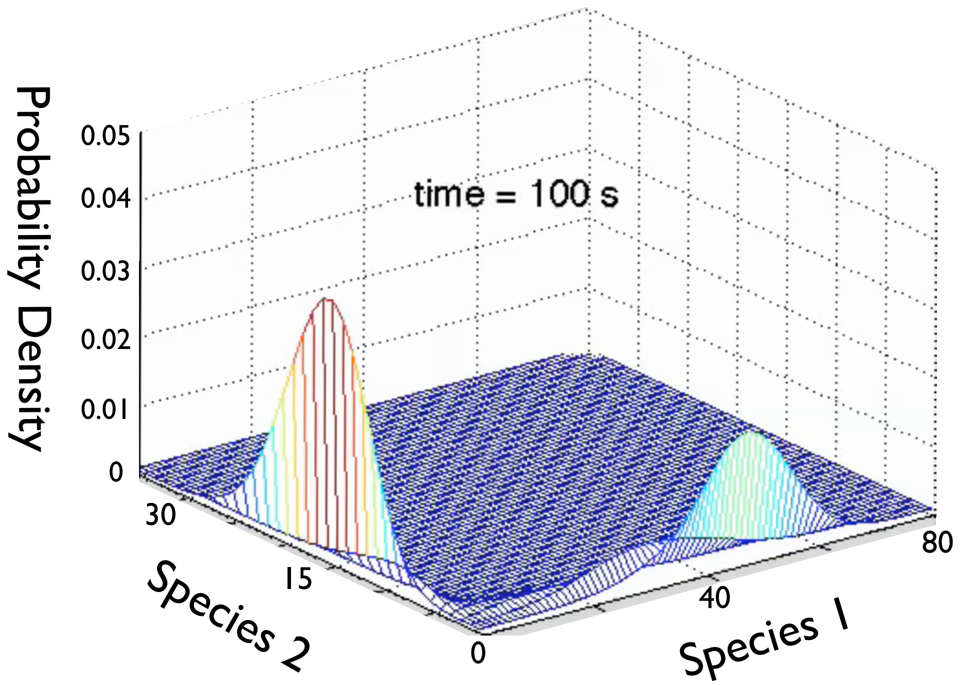
$$a_4(u,v) = v \qquad \nu_4 = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$$

$$\alpha_1 = 50 \qquad \beta = 2.5$$

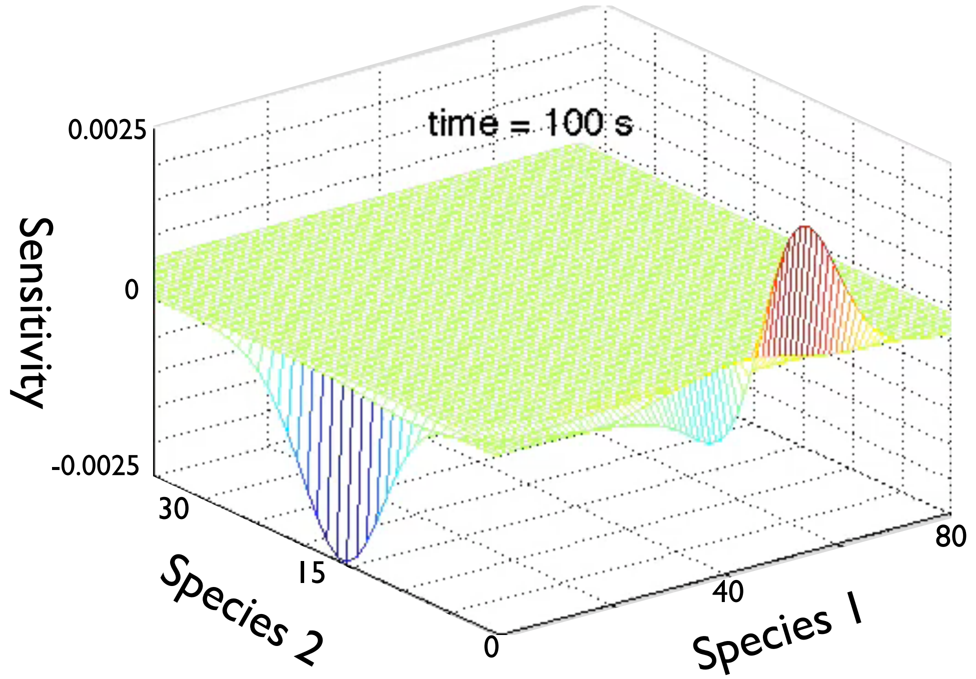
$$\alpha_2 = 16 \qquad \gamma = 1$$

Sensitivity Analysis

The precision of the FSP allows for accurate sensitivity analyses.



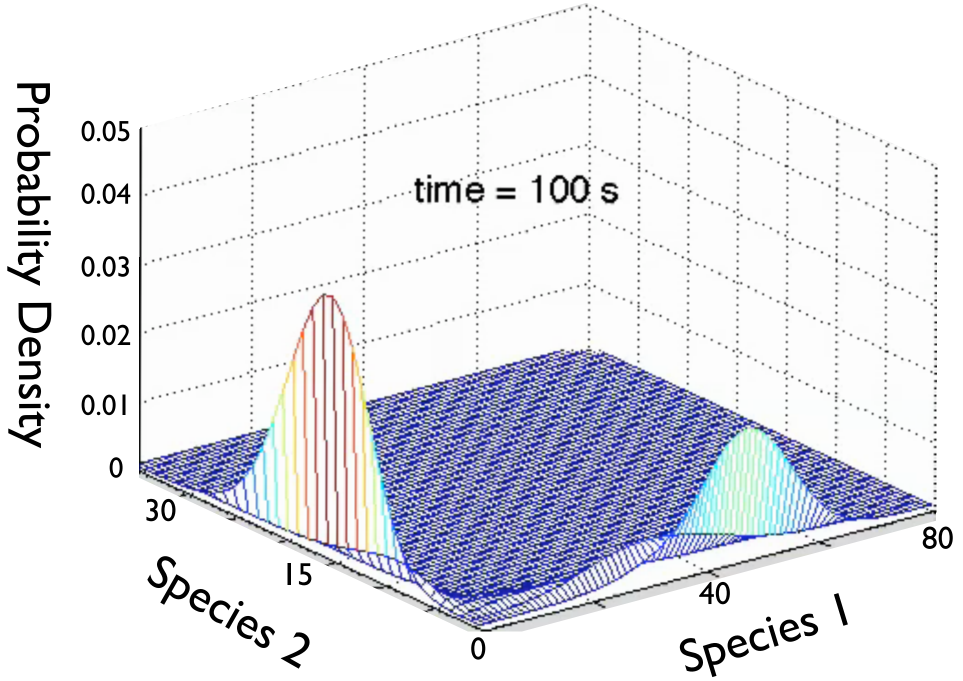
Nominal Distribution



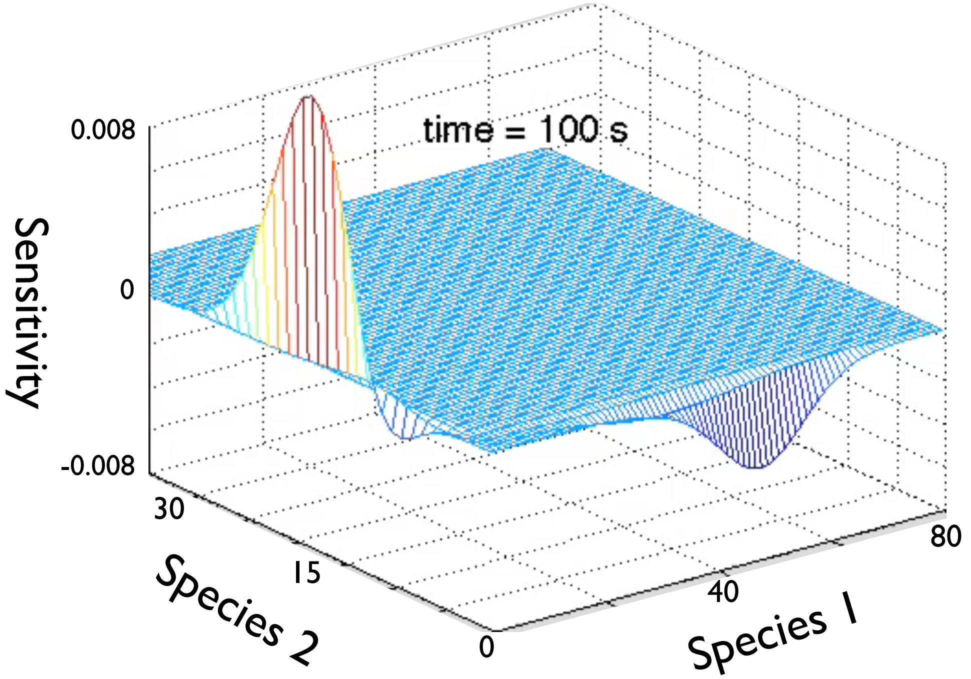
Sensitivity to α_1

Sensitivity Analysis

The precision of the FSP allows for accurate sensitivity analyses.



Nominal Distribution.



Sensitivity to α_2

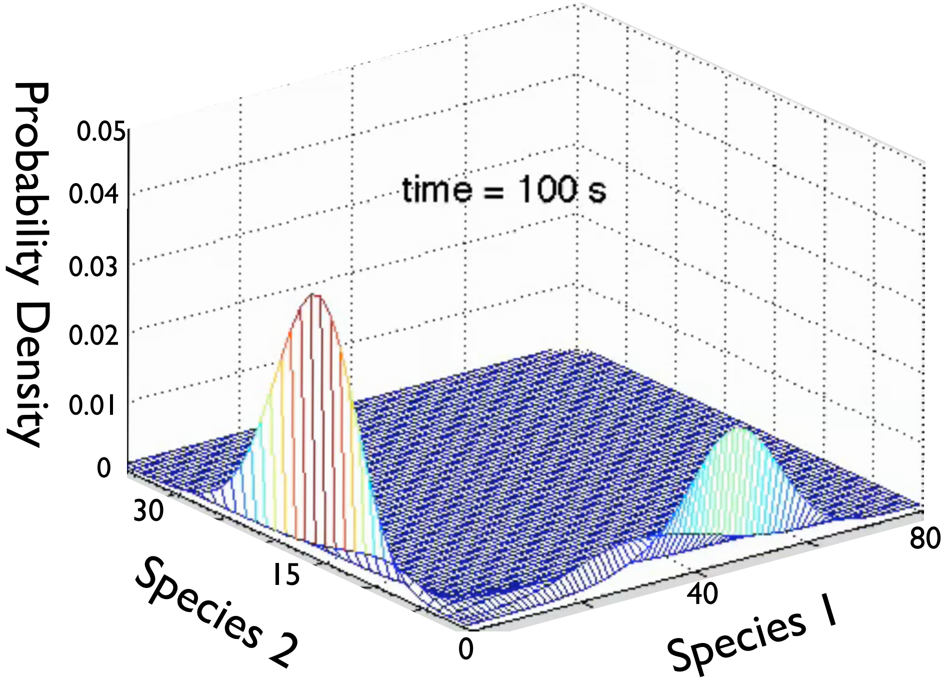
Sensitivity Analysis

Brian Munsky

The precision of the FSP allows for accurate sensitivity analyses.

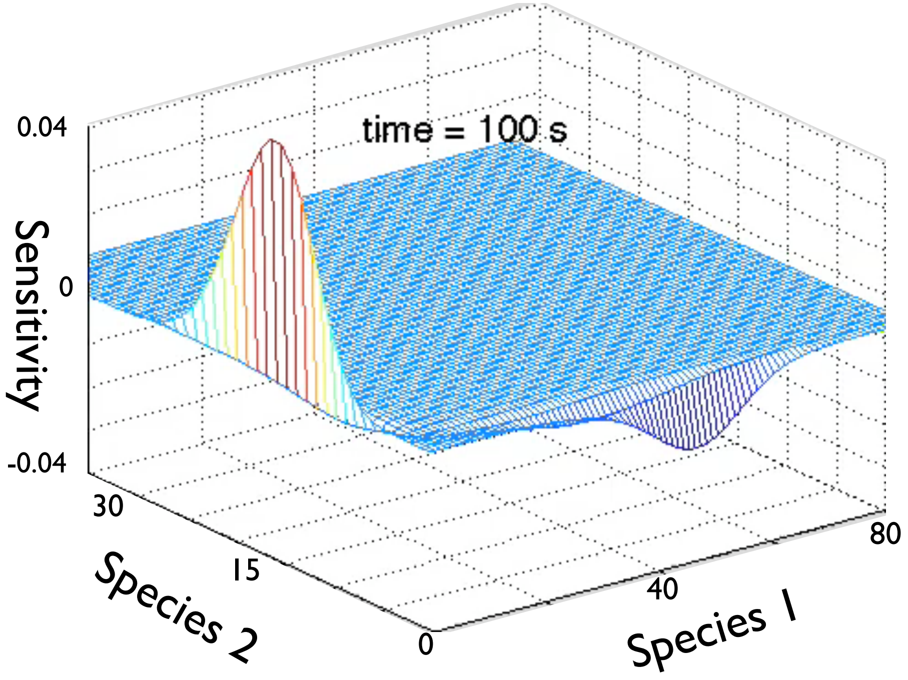
Production of *u:*

$$a_1(u,v) = \frac{\alpha_1}{1+v^\beta}$$



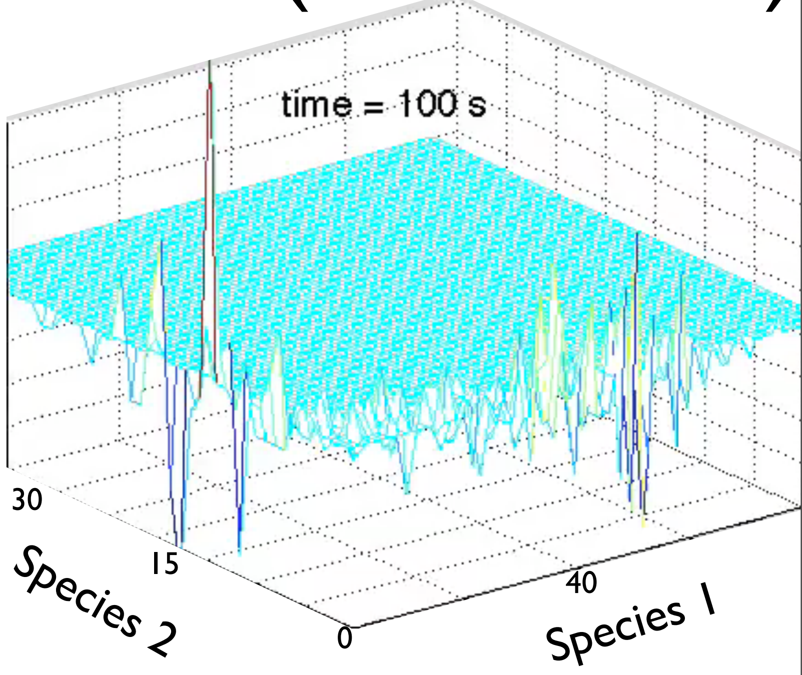
Nominal Distribution

FSP



Sensitivity to β
(FSP - 10s)

SSA (1000 runs)



Sensitivity to β
(SSA - 216s)

Identifying Toggle Parameters

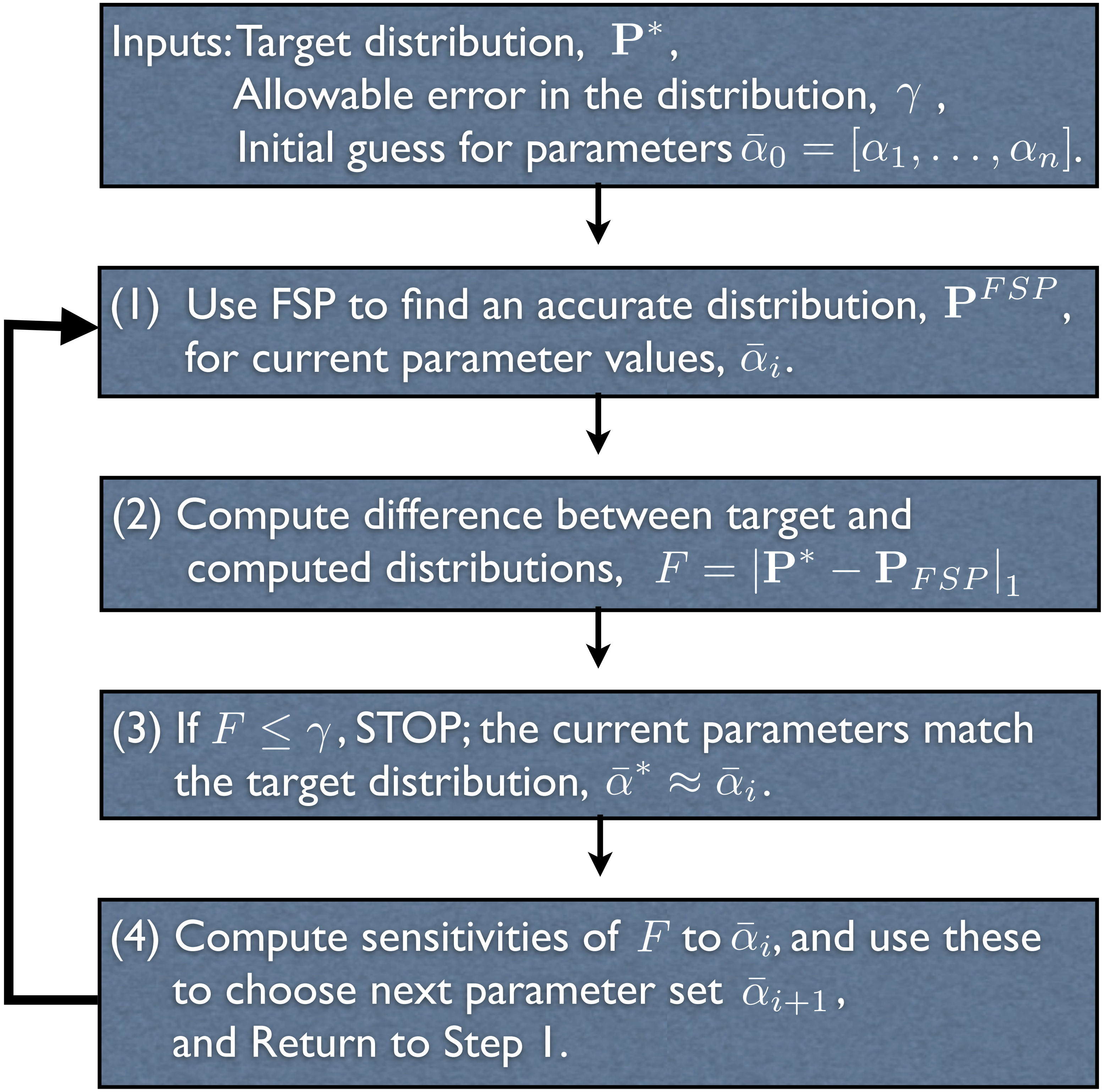
Brian Munsky

Most biological parameters are poorly known and difficult to measure.

By providing efficient and precise solutions for the CME, the FSP may help to systematically identify these parameters.

The target function, $\mathbf{P}^*$, can come from experimental observations or from more complex models.

The objective function, F, can be altered to emphasize the importance of different aspects of the distribution.



Inputs: Target distribution, $\mathbf{P}^*$,
Allowable error in the distribution, γ ,
Initial guess for parameters $\bar{\alpha}_0 = [\alpha_1, \ldots, \alpha_n]$.

(1) Use FSP to find an accurate distribution, $\mathbf{P}^{FSP}$, for current parameter values, $\bar{\alpha}_i$.

(2) Compute difference between target and computed distributions, $F = |\mathbf{P}^* - \mathbf{P}_{FSP}|_1$

(3) If $F \leq \gamma$, STOP; the current parameters match the target distribution, $\bar{\alpha}^* \approx \bar{\alpha}_i$.

(4) Compute sensitivities of F to $\bar{\alpha}_i$, and use these to choose next parameter set $\bar{\alpha}_{i+1}$, and Return to Step 1.

Identifying Toggle Parameters

Brian Munsky

Actual Parameters (unknown):

$$\alpha^* = [\alpha_1, \alpha_2, \beta] = [50, 16, 2.5]$$

Initial guess:

$$\alpha_0 = [40, 20, 1.5]$$

Initial Error:

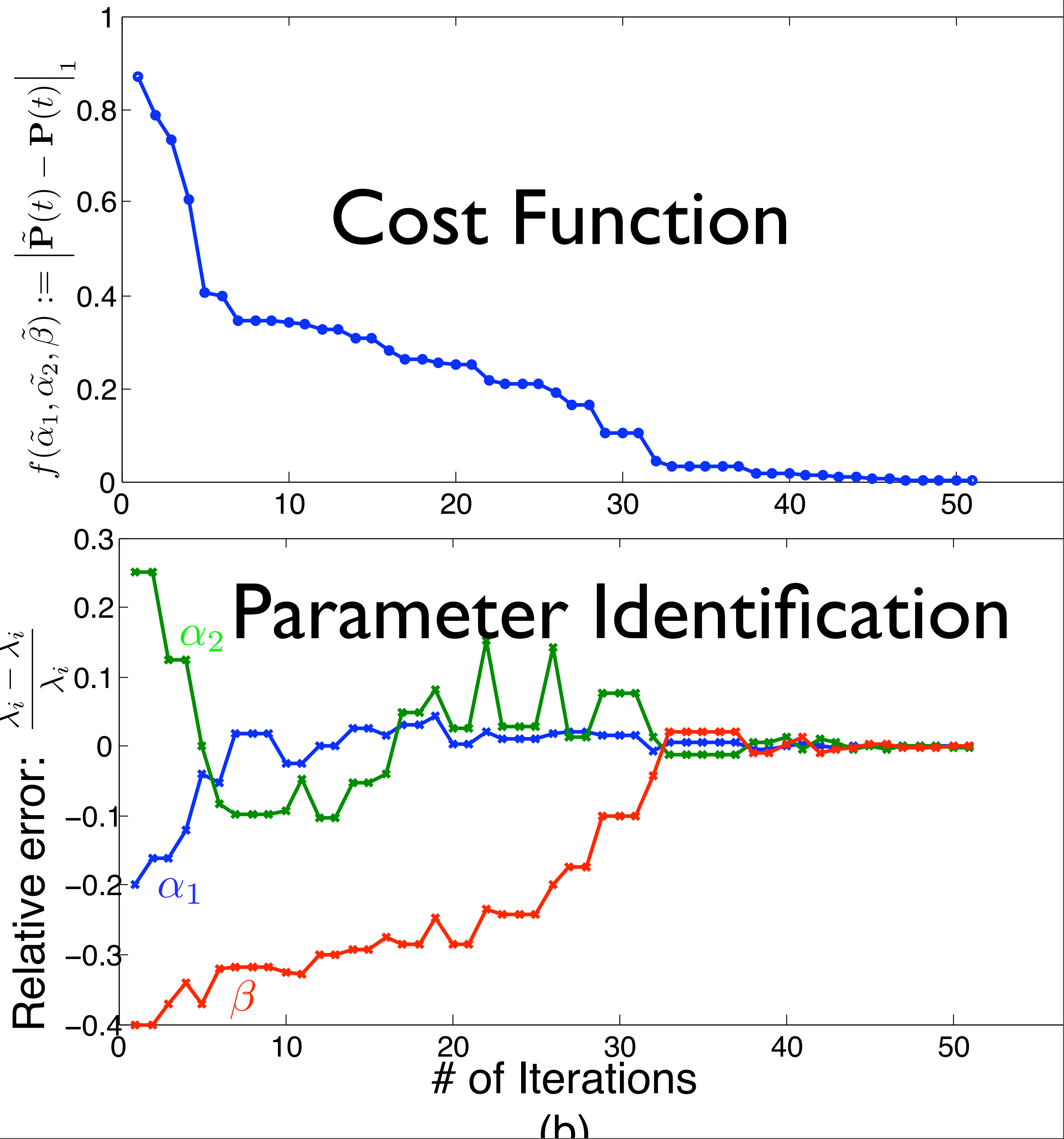
$$F_0 = ||\mathbf{P}(\alpha_0) - \mathbf{P}(\alpha^*)||_1 \approx 0.9$$

Parameters after 50 iterations:

$$\alpha_{50} = [49.99, 15.97, 2.504]$$

Error after 50 iterations

$$F_{50} \approx 0.02$$



Conclusions

- Stochastic fluctuations or “noise” is present in the cell
  - Random nature of reactions
  - Quantization of reactants
  - Low copy numbers
- Fluctuations may be very important
  - Cell variability
  - Cell fate decisions
- Some tools are available
  - Monte Carlo simulations (SSA and variants)
  - Moment approximation methods
  - Linear noise approximation (Van Kampen)
  - Finite State Projection
- Many more are needed!

Conclusions

The Finite State Projection: a new tool for stochastic analysis of gene networks

Advantages:

- Accuracy: solutions with a guaranteed error bounds
  Particularly suitable for studying rare events
- Speed: solutions can be much faster than Monte Carlo methods
  especially when the system has large number of reactions/reaction firings
- Insight: Provides valuable information at little additional cost:
  Sensitivity/robustness to parameter changes
  Effect of changes in initial probabilities

Limitations

- Scalability: Not feasible when there are many species with broad distributions (over the time of interest [0, t])