

# Stochastic Gene Expression in Systems Biology (Part 2)

---

**Brian Munsky**

Center for Non-Linear Studies, Los Alamos National Lab

# On the menu...

---

- Monday

- Overview of Stochastic Gene Expression
- Stochastic Chemical Kinetics
- Solutions for Simple Stochastic Processes (Transcription)
- Importance of Population Size
- Moment Computations for Linear Propensities
- Linear Noise Approximation

- Today

- ▶ Monte Carlo Simulation Techniques
  - \* Gillespie (SSA), Tau leaping, Chemical Langevin (SDEs), Slow Scale SSA.
- ▶ Density Computations with Finite State Projection Techniques
- ▶ Switch and Trajectory Analyses



Brian Munsky

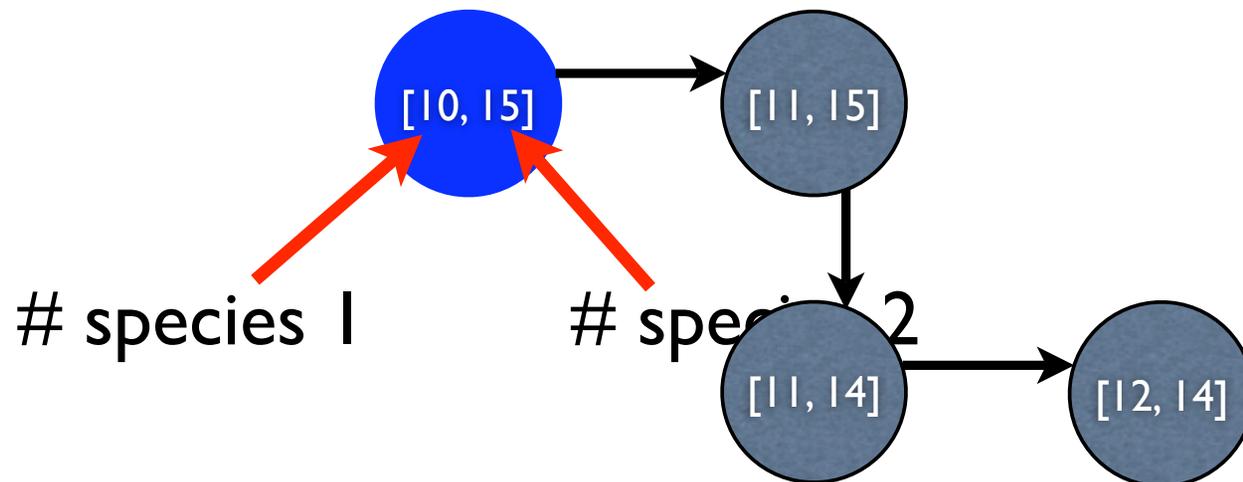
# Kinetic Monte-Carlo Simulation Methods

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

---

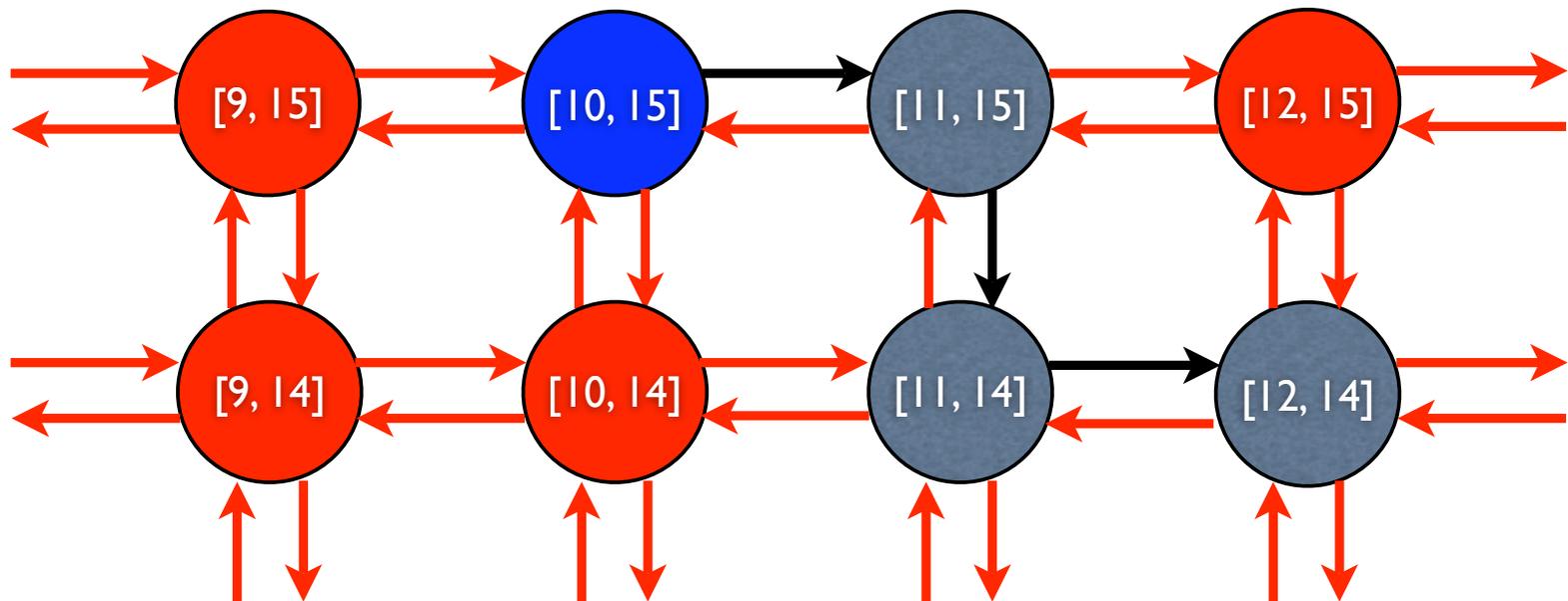
- At any time, the state of the system is defined by its integer population vector:  $\mathbf{x} \in \mathbb{Z}^N$
- Reactions are transitions from one state to another:



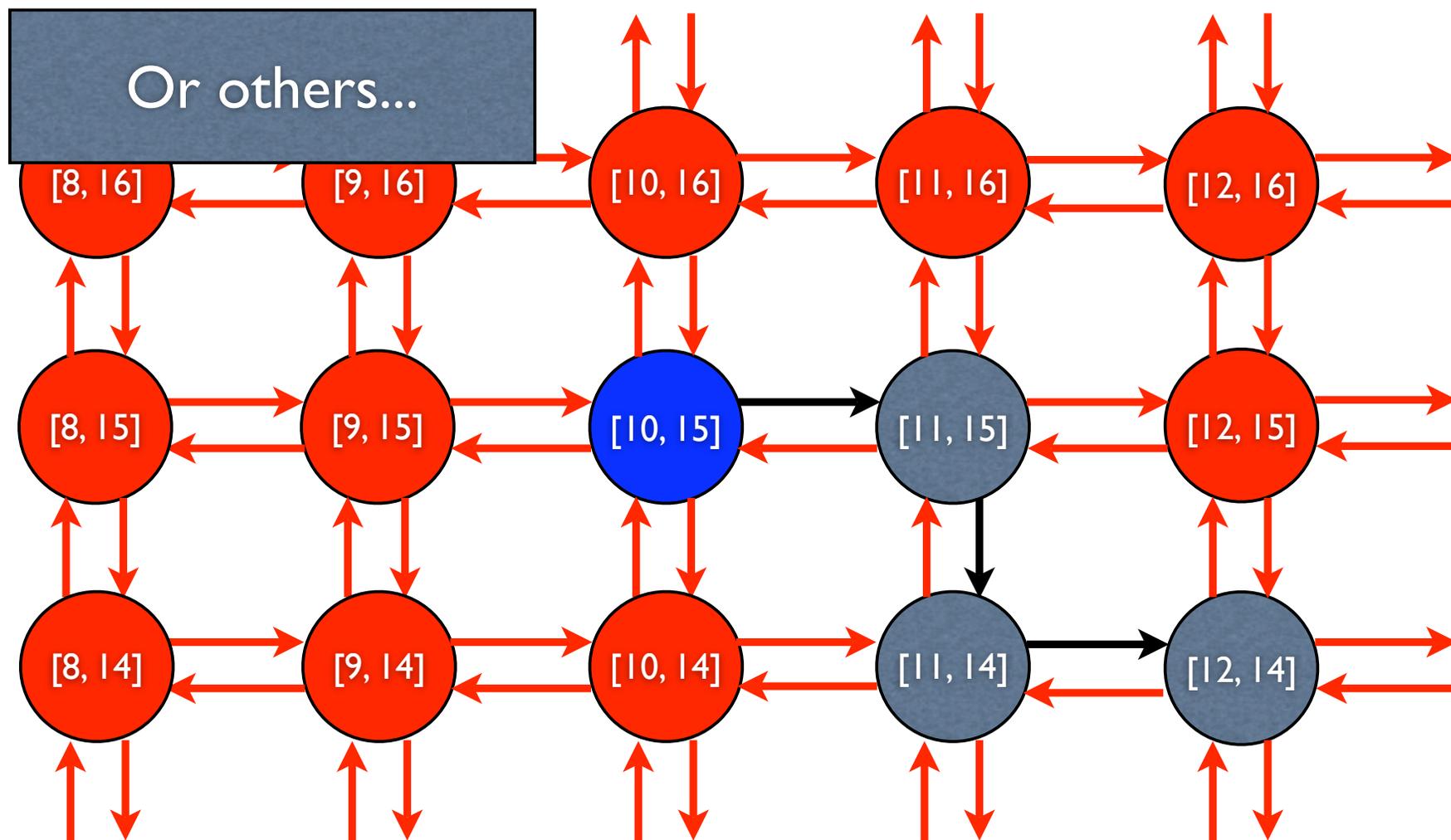
# A Jump-Markov description of chemical kinetics

---

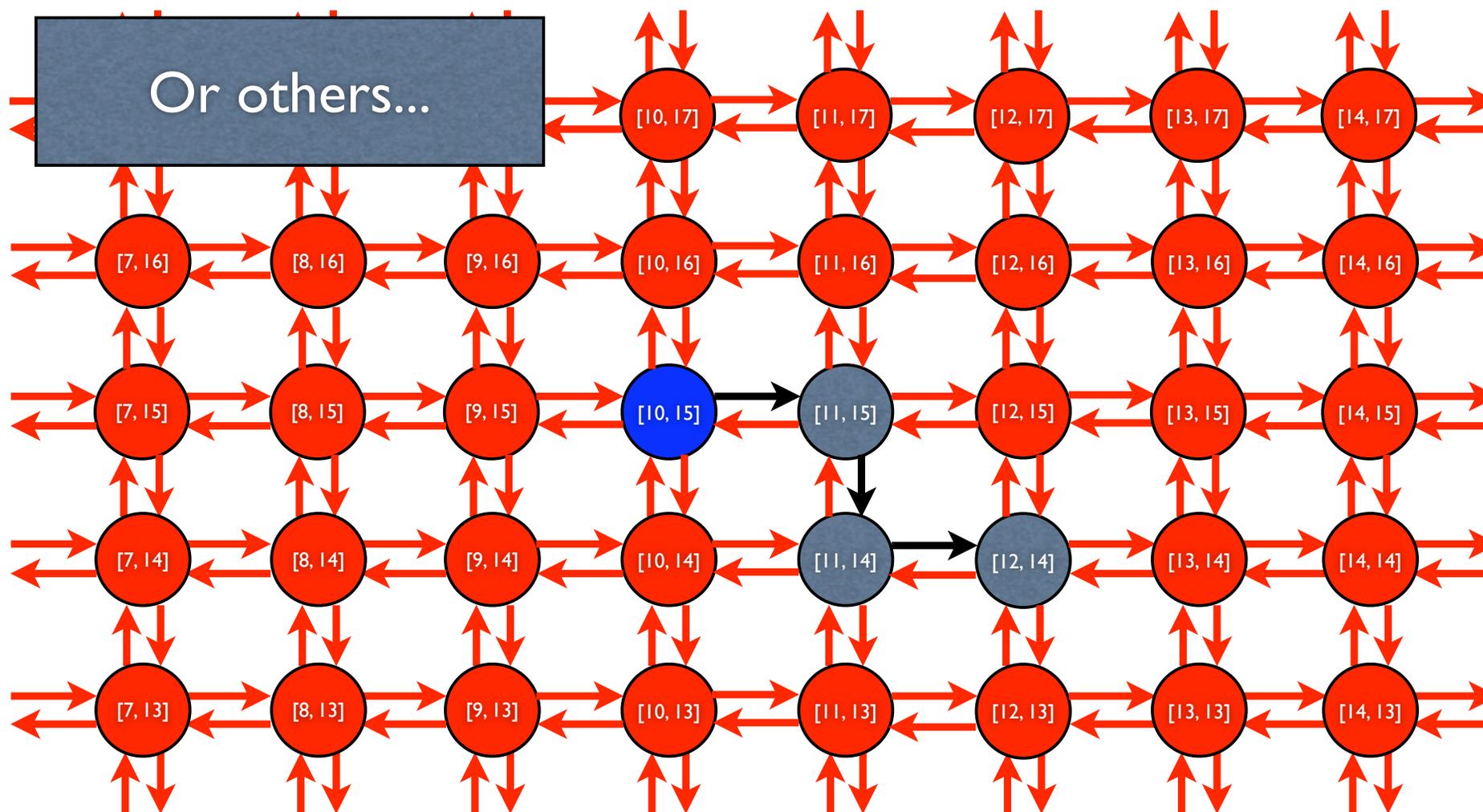
- At any time, the state of the system is defined by its integer population vector:  $\mathbf{x} \in \mathbb{Z}^N$
- Reactions are transitions from one state to another:
- These reactions are random, others could have occurred:



# A Jump-Markov description of chemical kinetics

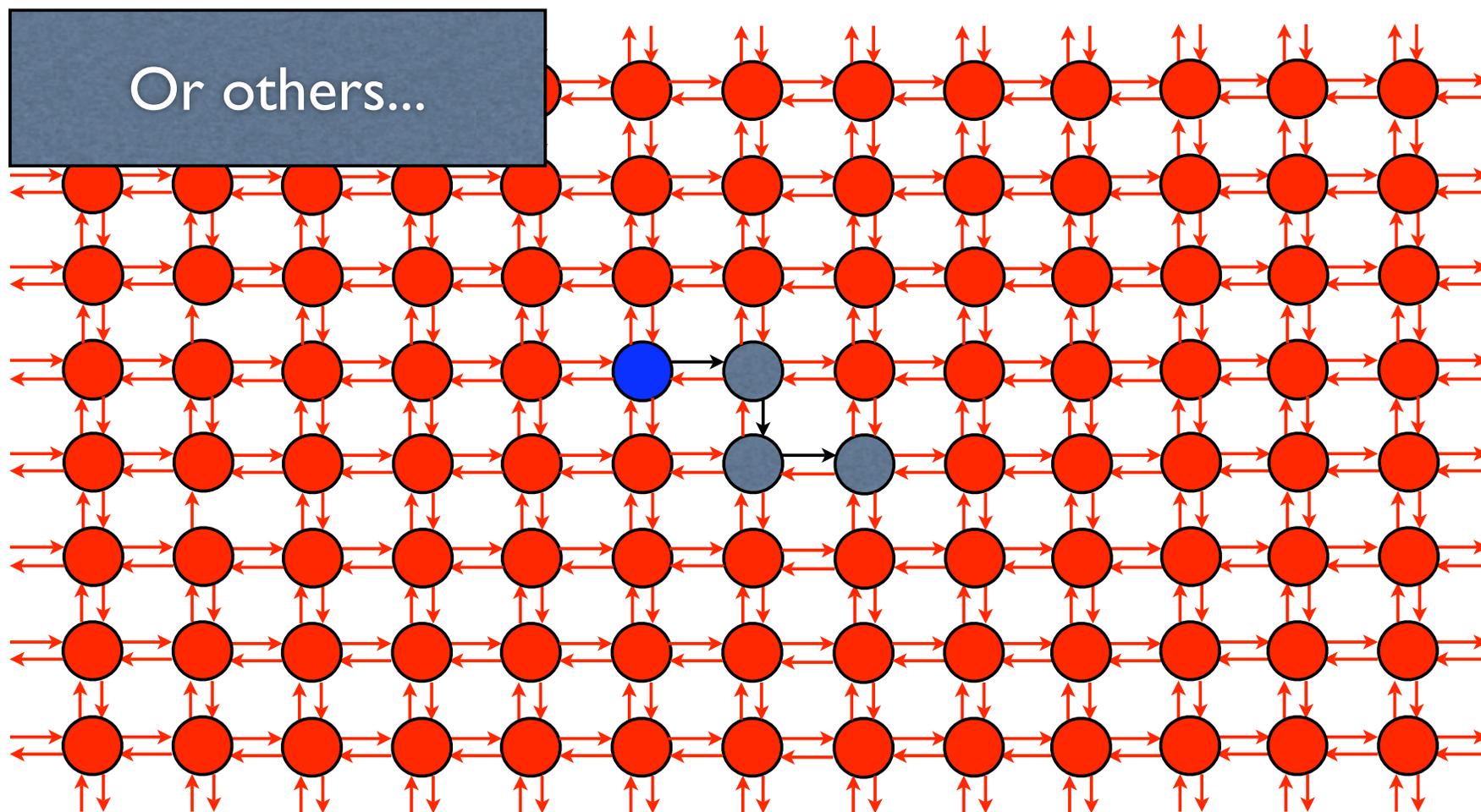


# A Jump-Markov description of chemical kinetics



# A Jump-Markov description of chemical kinetics

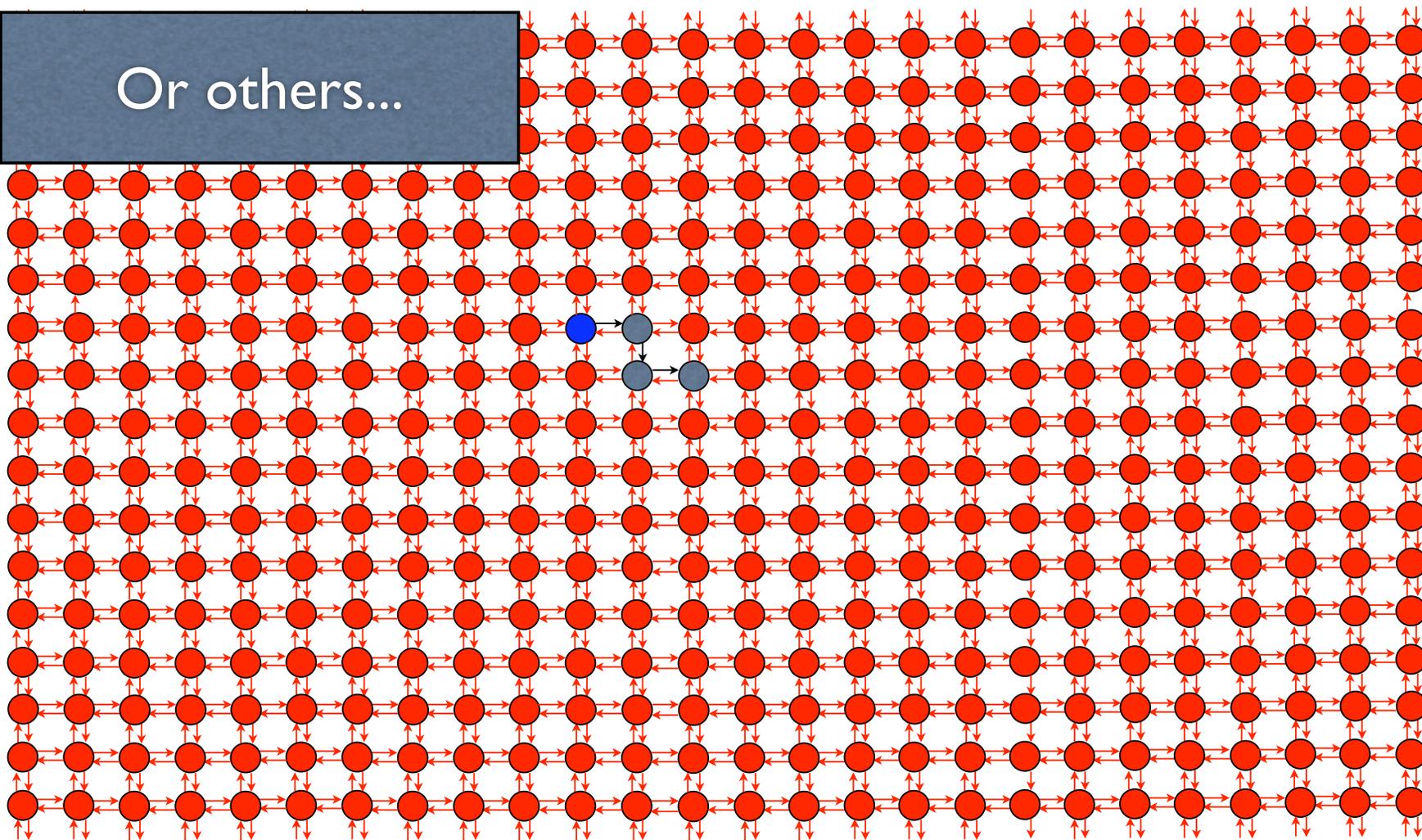
---



# A Jump-Markov description of chemical kinetics

---

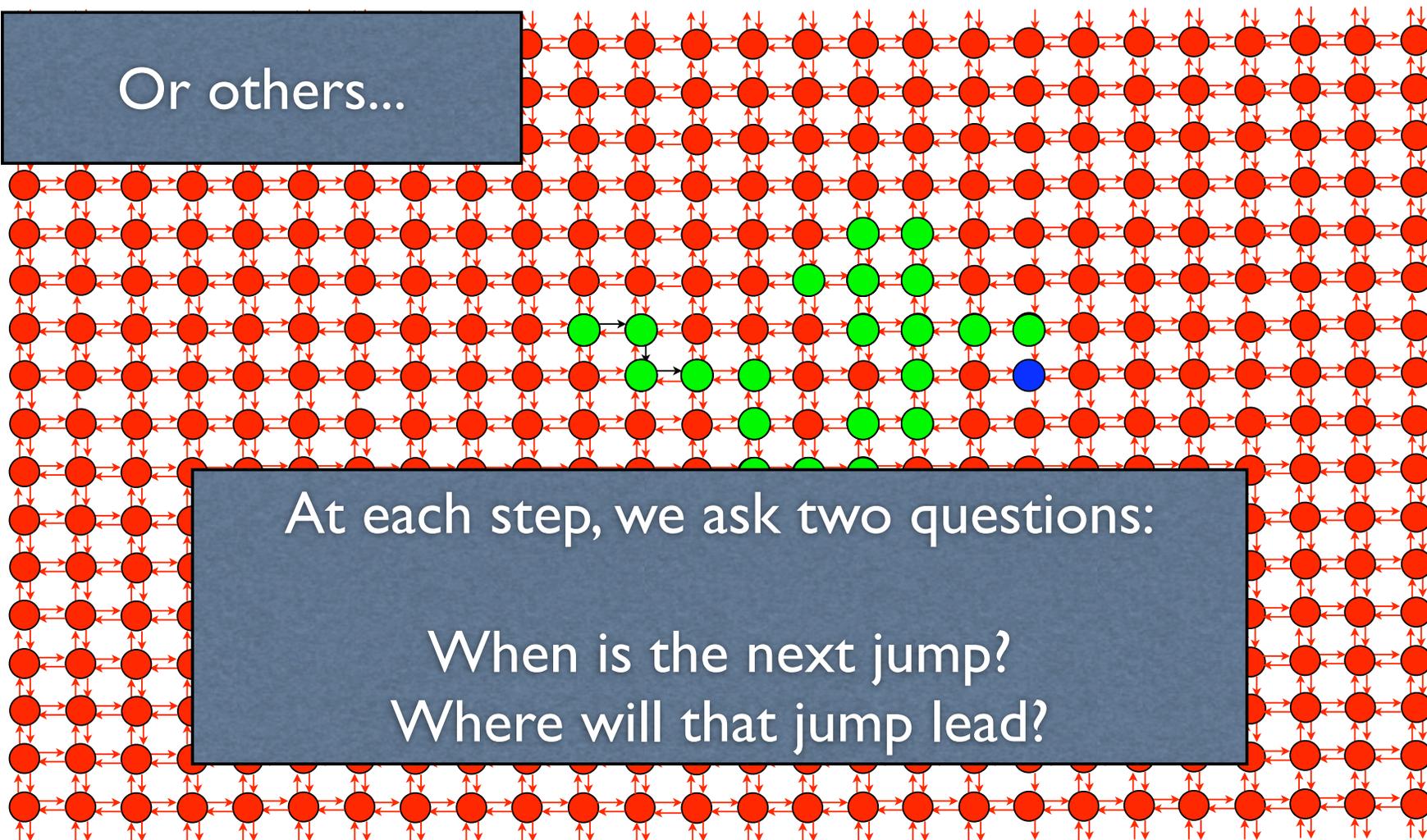
Or others...



# A Jump-Markov description of chemical kinetics

---

Or others...



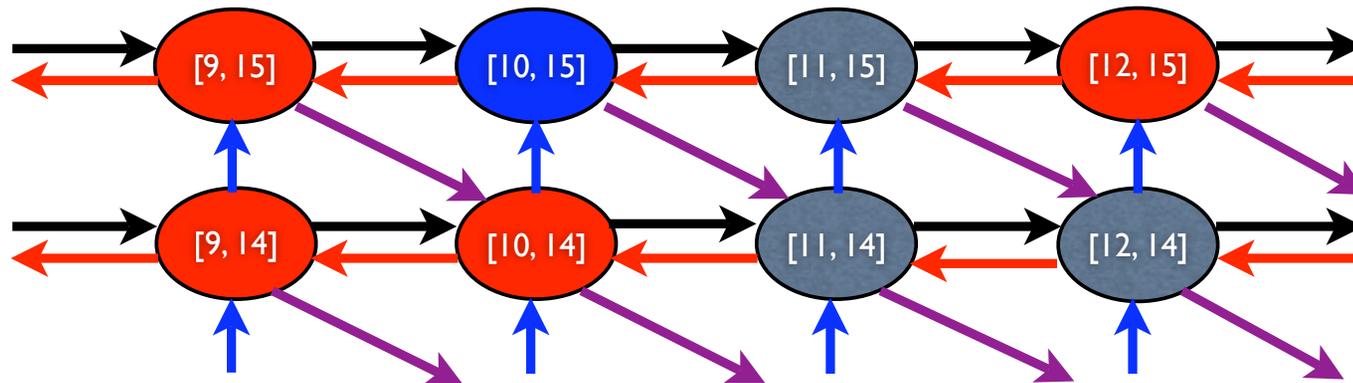
At each step, we ask two questions:

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

# Reaction Stoichiometry (review)

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

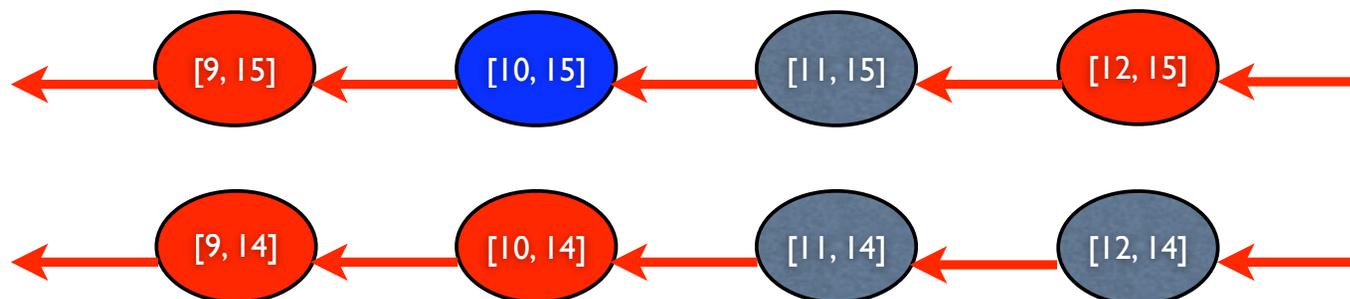
$\mathbf{s}_1 = [1, 0]^T$ $\mathcal{S}_1 \rightarrow \mathcal{S}_1 + \mathcal{S}_1$ $\mathcal{S}_2 \rightarrow \mathcal{S}_2 + \mathcal{S}_1$ $\emptyset \rightarrow \mathcal{S}_1$	$\mathbf{s}_2 = [-1, 0]^T$ $\mathcal{S}_1 + \mathcal{S}_1 \rightarrow \mathcal{S}_1$ $\mathcal{S}_1 + \mathcal{S}_2 \rightarrow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



# Reaction Propensities (review)

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

$s_2 = [-1, 0]^T$	$w_2(x_1, x_2)$
$\mathcal{S}_1 + \mathcal{S}_1 \rightarrow \mathcal{S}_1$	$k_1 x_2 (x_1 - 1) / 2$
$\mathcal{S}_1 + \mathcal{S}_2 \rightarrow \mathcal{S}_2$	$k_2 x_1 x_2$
$\mathcal{S}_1 \rightarrow \emptyset$	$k_3 x_1$



# 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  $[t, t + 2\Delta t)$ :  $(1 - w\Delta t + \mathcal{O}(\Delta t)^2)^2 = 1 - 2w\Delta t + \mathcal{O}(\Delta t)^2$

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

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

# 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:

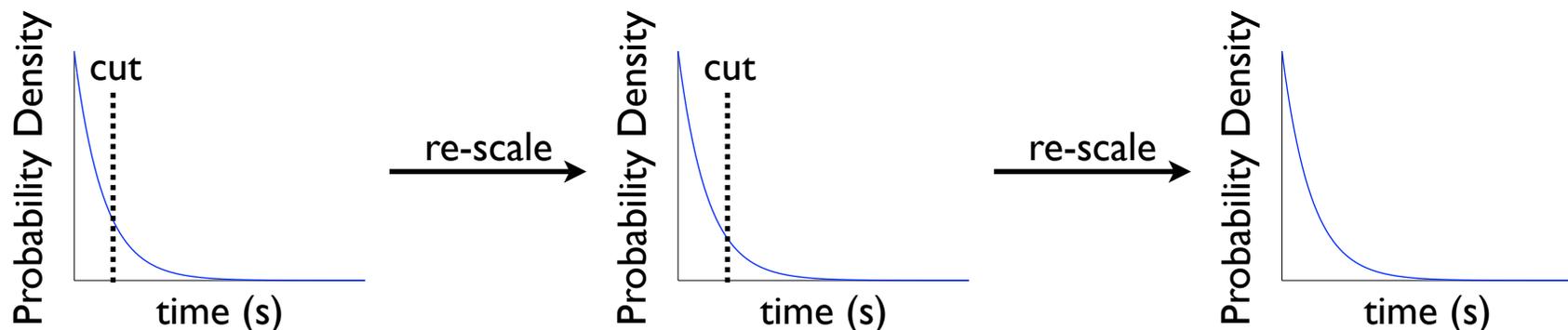
$$f_T(\tau) = \frac{1}{w} \exp(-w\tau)$$

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

Notation:  $T \in \text{EXP}(\lambda) \rightarrow T$  is an exponentially distributed r.v. with parameter:  $\lambda$ .

# 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 look 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,

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

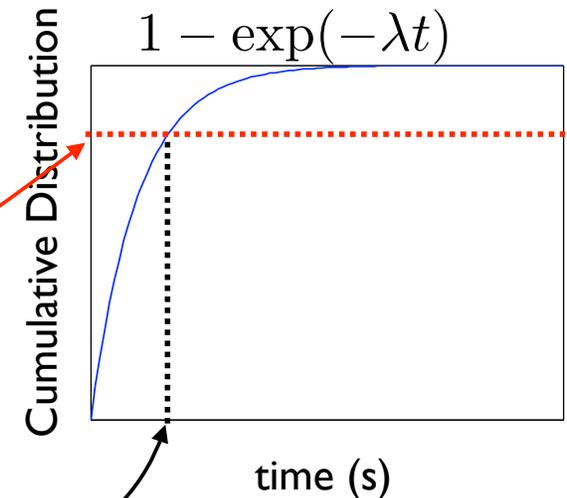
- Generate uniform random number,

$$r \in U[0, 1]$$

- 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

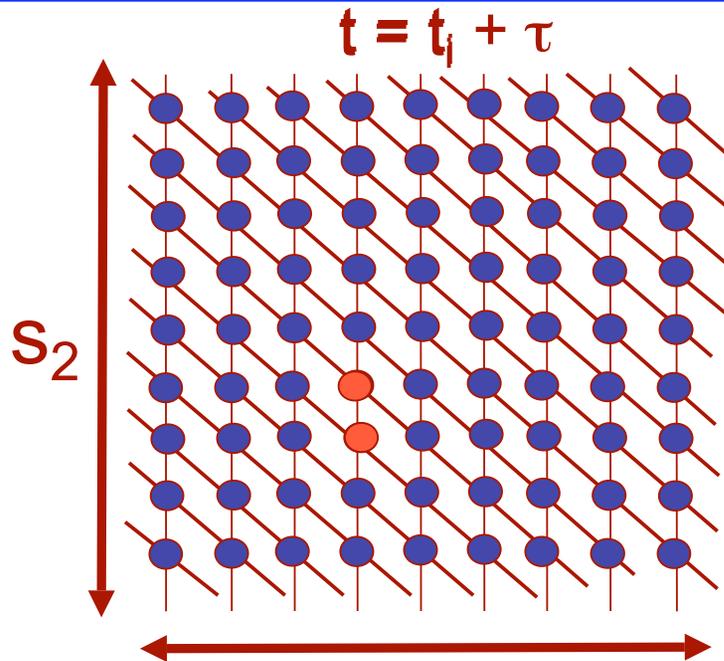
---

## 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.

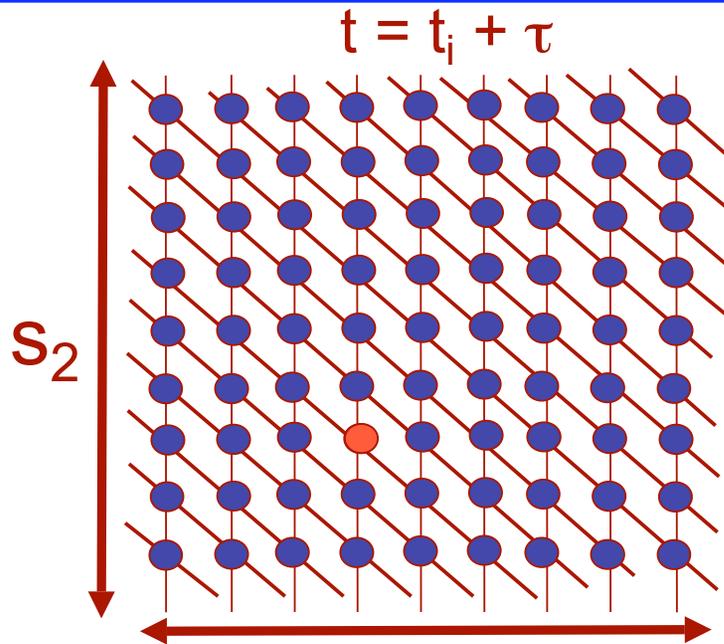
Step 3. Update current Time ( $t=t+\tau$ ) and State ( $\mathbf{x} = \mathbf{x}+s_k$ ).

# 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  $\tau_{\mu}$ :

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

Step 3. Update current Time ( $t=t+\tau_k$ ) and State ( $\mathbf{x} = \mathbf{x}+s_k$ ).

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,  $\tau_k$ .
- 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  $\{\tau_1, \tau_2, \dots, \tau_M\}$  be a set of exponentially distributed random variables:  $\tau_\mu \in \text{EXP}(w_\mu)$

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

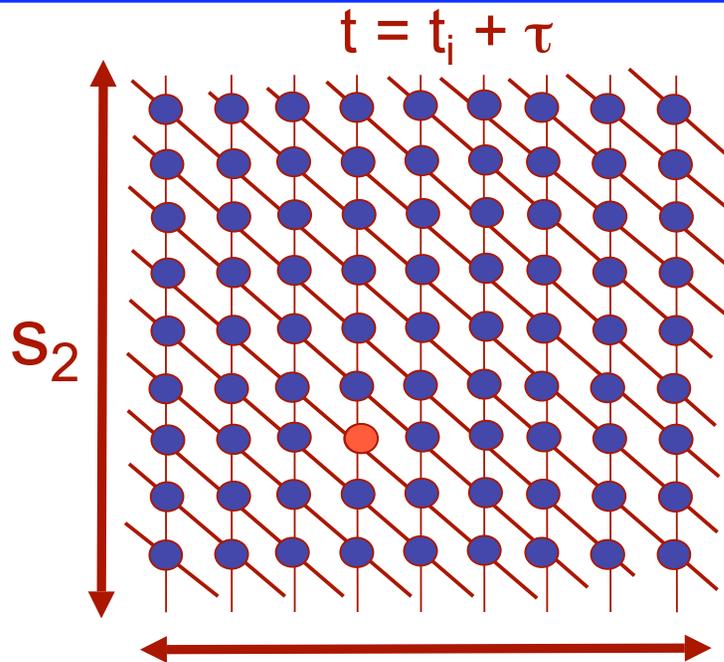
$$\min_{\mu \in \{0, \dots, M\}} \tau_\mu \in \text{EXP}(|\mathbf{w}|_1)$$

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:

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

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 ( $\mathbf{x} = \mathbf{x}+s_k$ ).

# 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).
- $\tau$ -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)

# $\tau$ Leaping

---

Step 0. Specify length of each time step,  $\tau$ .

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

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

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

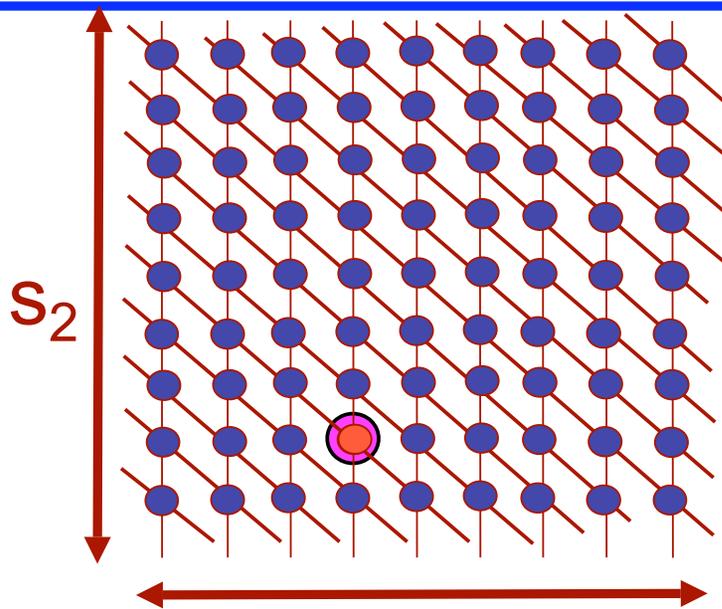
Step 1. For each  $\mu$ , generate  $k_\mu$ .

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)

# $\tau$ Leaping



$t = t_i + \tau$  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  $\mathbf{w}_\mu \tau$ :  $P_{k_\mu}(n) = \frac{[w_\mu(\mathbf{x})\tau]^n}{n!} e^{-w_\mu(\mathbf{x})\tau}$

Step 1. For each  $\mu$ , generate  $k_\mu$ .

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 $\tau$ leaping

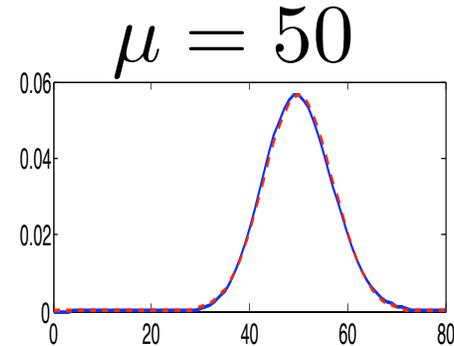
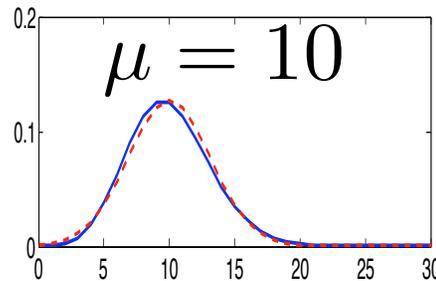
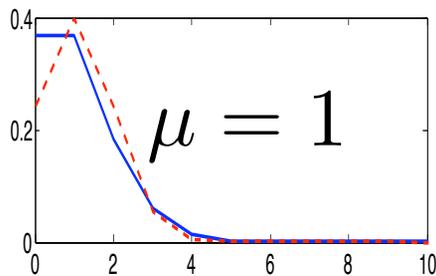
---

- For many situations  $\tau$  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.

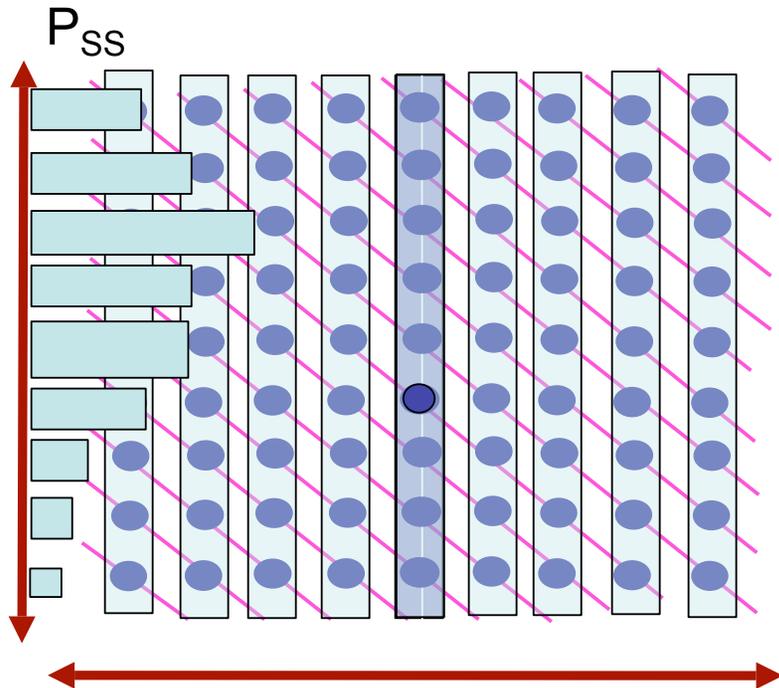
# Monte-Carlo Simulation Methods

---

- Stochastic Simulation Algorithm (SSA).
- $\tau$ -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.

---



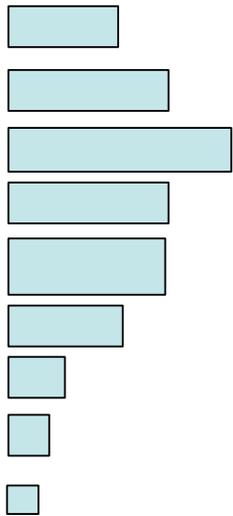
Separate into “fast” and “slow” partitions.

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

# Fast--Slow partitions.

---

$P_{ss}$



Slow Reaction  
Propensities

$$\times \begin{bmatrix} w_{\mu}(\mathbf{x}_1) \\ w_{\mu}(\mathbf{x}_2) \\ w_{\mu}(\mathbf{x}_3) \\ \vdots \end{bmatrix}$$

Average Slow  
Reaction Propensities

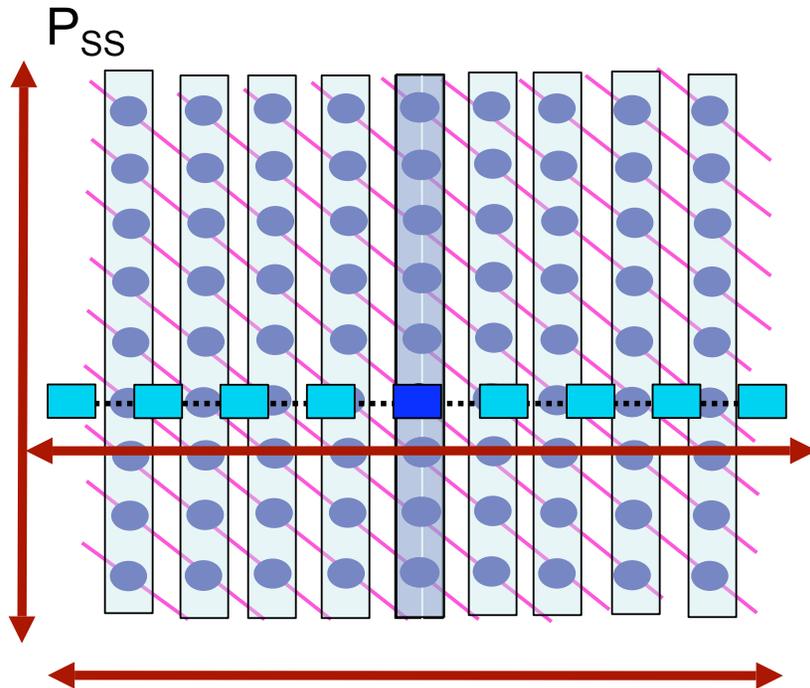
$$= \bar{w}_{\mu}, \text{ for } \mu = \{1, 2, \dots, M\}$$

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.

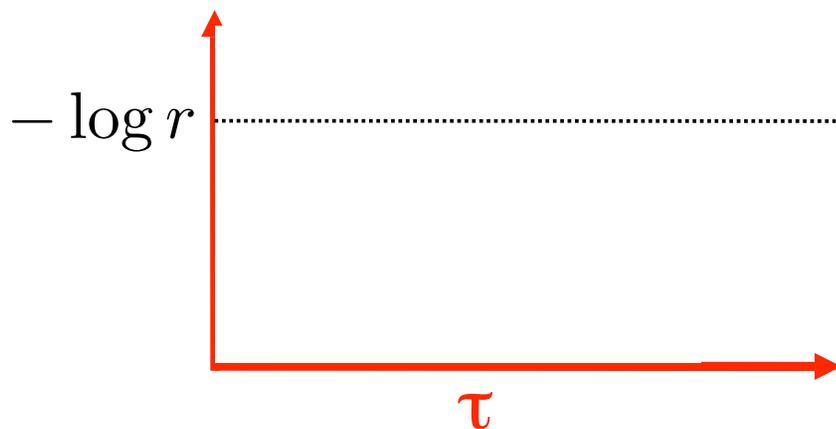
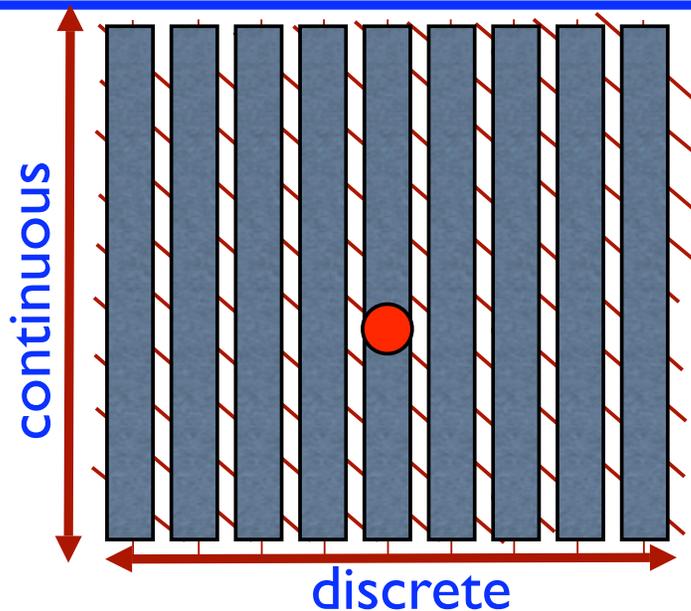
This is simulated with SSA.

## 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,  $r$ .

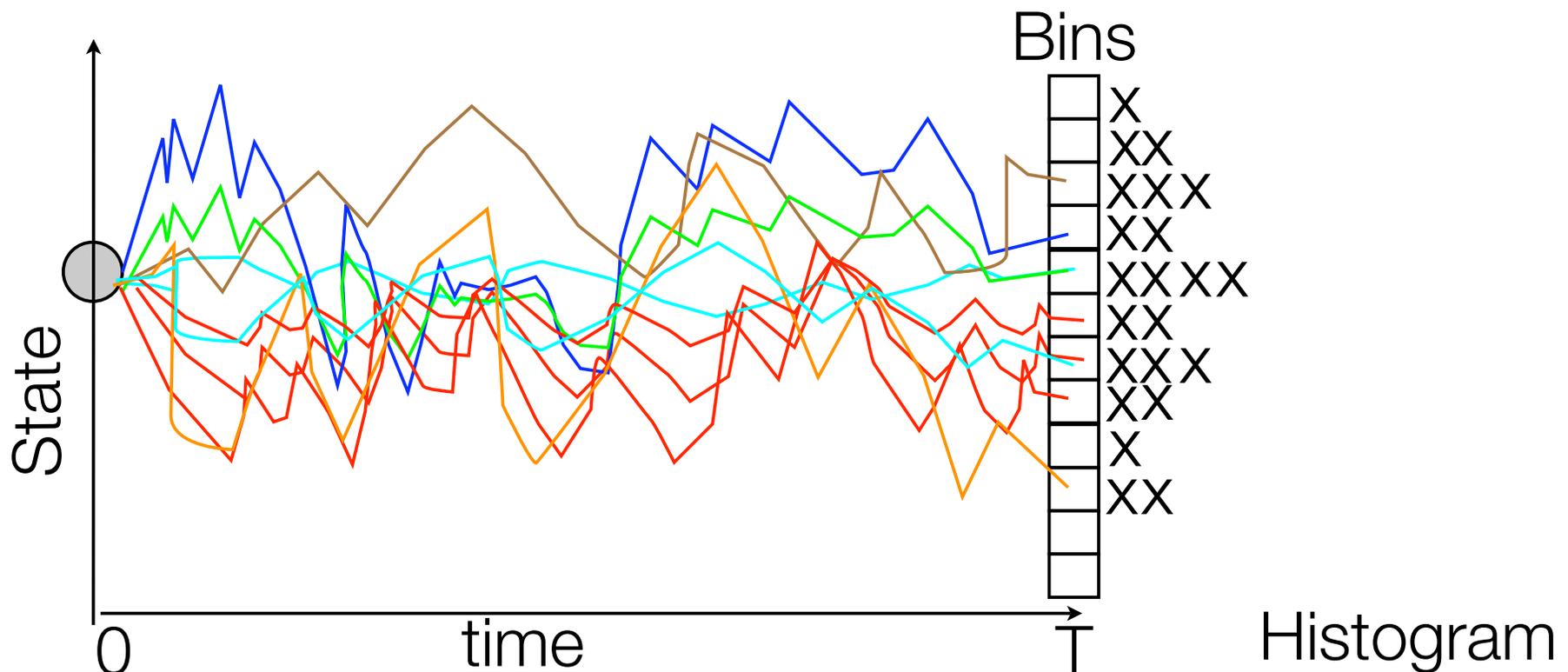
Numerically integrate propensity functions until:

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

Choose next discrete reaction.

# Using the SSA to Find Distributions

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



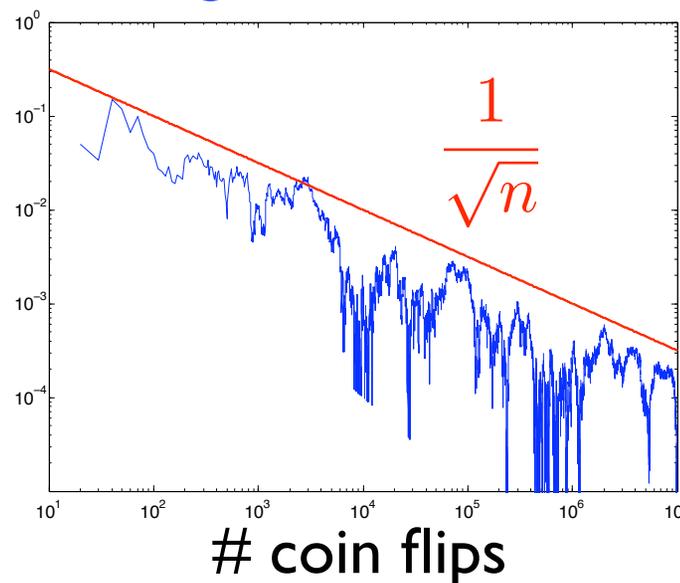
# Convergence of the SSA

- 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  $3 \times 10^{-4}$ .

# Density Computations

---

● The Finite State Projection (FSP) solution to the Chemical Master Equation.

● Reductions to the FSP

● Case studies.

# The Chemical Master Equation

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

The CME (McQuarrie '67):

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

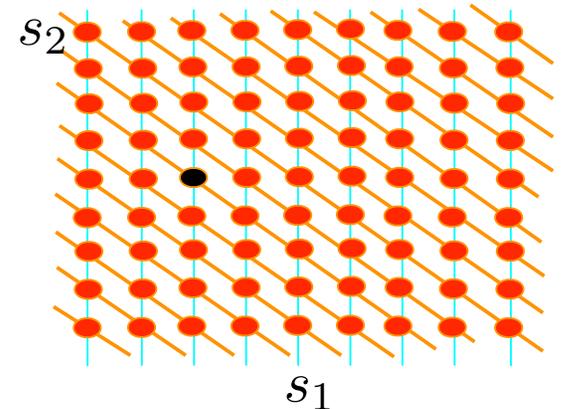
Define the probability density state

vector (pdv):  $\mathbf{P}(\mathbf{X}, t) := [p(\mathbf{x}_1, t), p(\mathbf{x}_2, t), p(\mathbf{x}_3, t), \dots]^T$ .

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

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

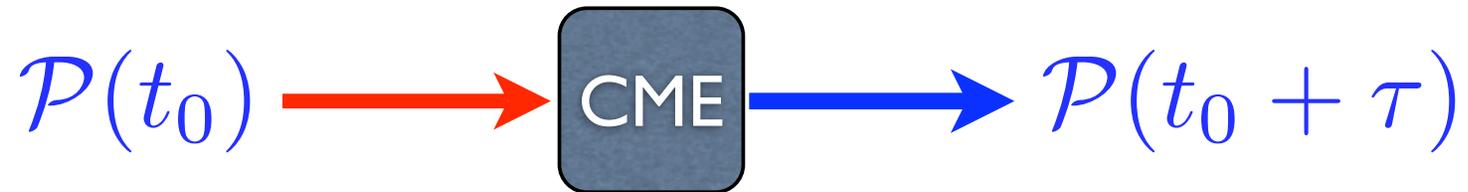
The matrix CME



# The Chemical Master Equation

---

- The solution of the CME is a transfer operator:



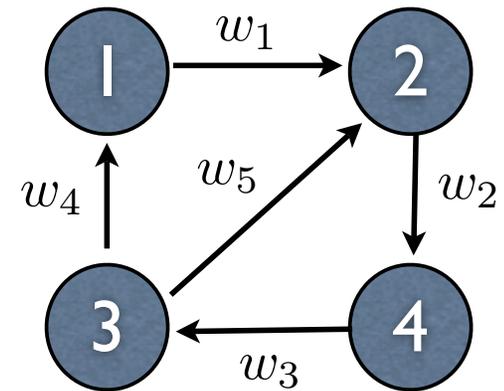
- The dimension of the CME can be INFINITE.
- Most CME's cannot be solved, so approximations are needed.

# 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  $A_{j,i}$



$$\mathbf{A} = \left[ \begin{array}{c} \\ \\ \\ \end{array} \right]$$

# The Finite State Projection

Select the states to keep.

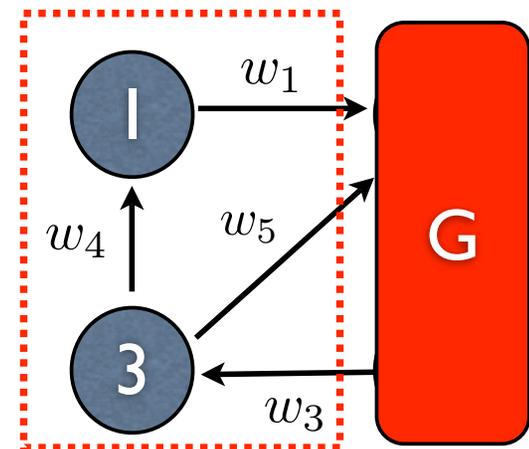
Find the corresponding projection matrix:

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

Collapse remaining states into a single absorbing

state

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

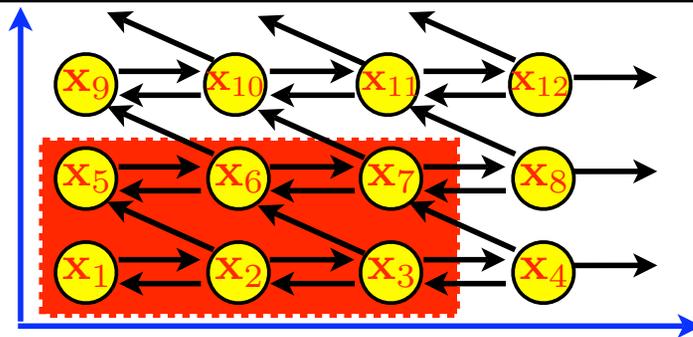


$$\mathbf{A} = \begin{bmatrix} -w_1 & 0 & w_4 & 0 \\ w_1 & -w_2 & w_5 & 0 \\ 0 & 0 & -w_4 - w_5 & w_3 \\ 0 & w_2 & 0 & -w_3 \end{bmatrix}$$

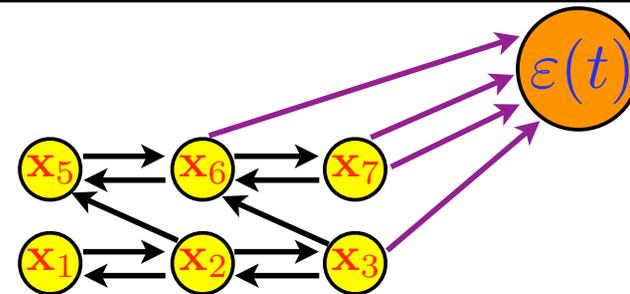
This is the generator for a new Markov chain

# 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 =  $\#(J) + \#(J') = \text{Infinite}$

FSP Master Equation

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

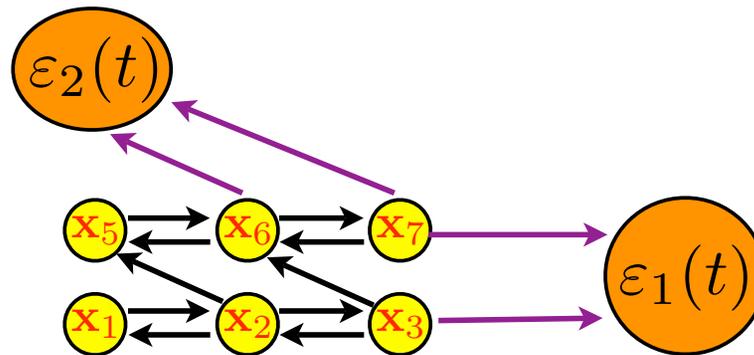
Dimension =  $\#(J) + 1 = 7$

The FSP Theorem  
(Munsky/Khammash JCP '06)

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

# A Test...

---



What do  $\varepsilon_1(t)$  and  $\varepsilon_2(t)$  mean?

# 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:

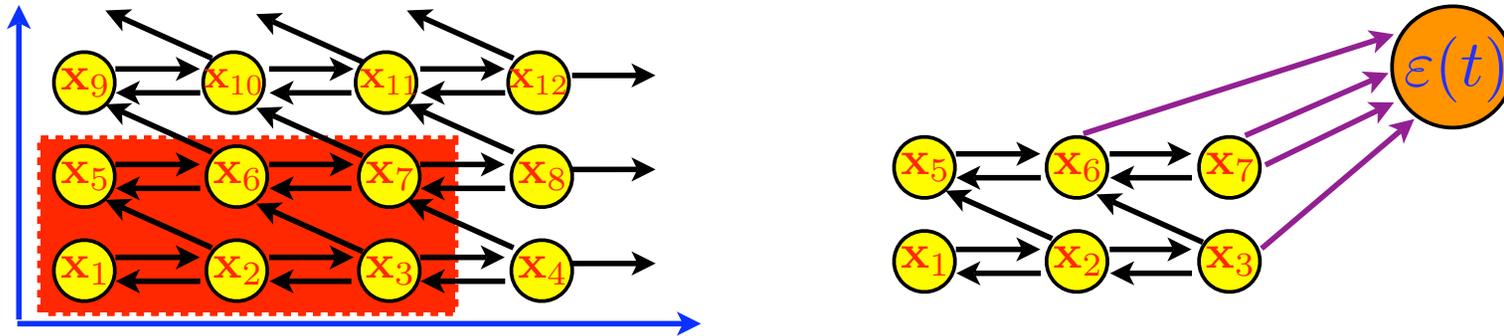
If  $\varepsilon_i(t_f) \leq \varepsilon_{\max}$ , **Stop**.

$\mathcal{P}_{J_i}^{FSP}(t_f)$  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.

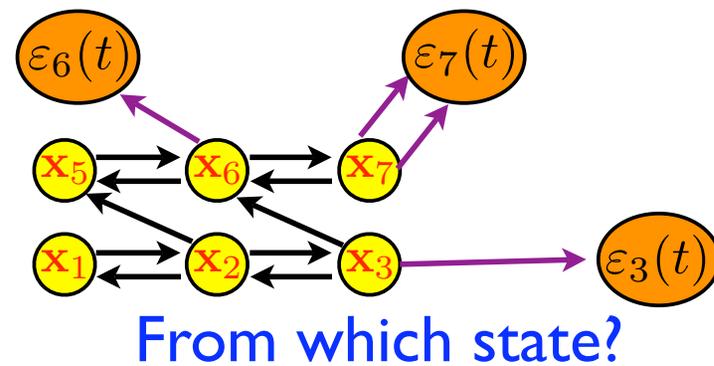
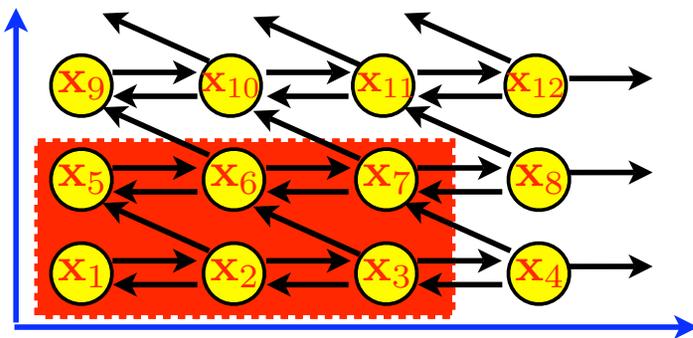
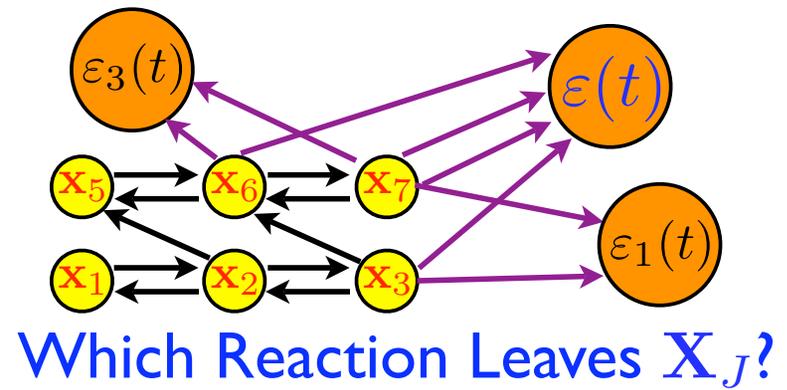
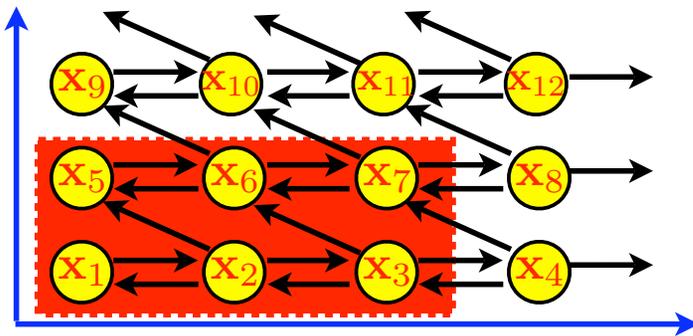
## The “error” sink of the FSP to get exit times.



- In the original FSP,  $\varepsilon(t)$  is the amount of the probability measure that exits the projection region  $X_J$ .
- 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  $X_J$ , but not how.

# Multiple FSP sinks to get exit directions.

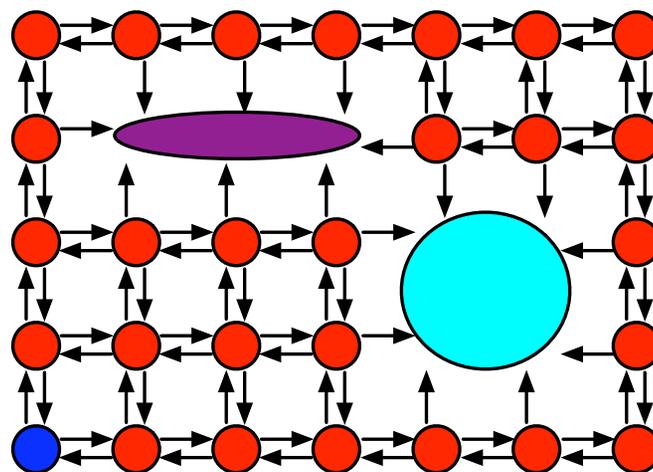
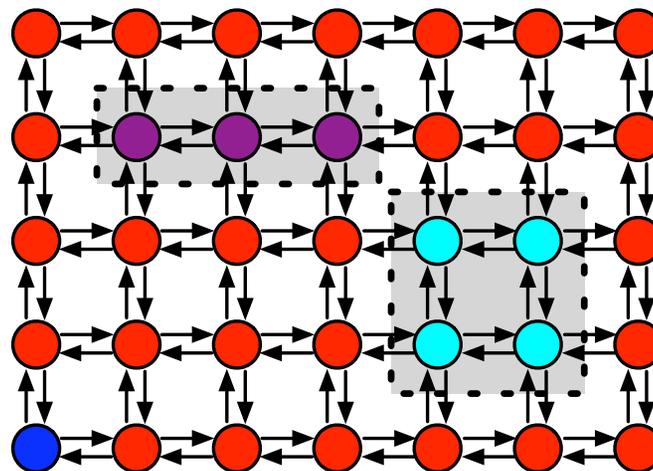
- By using multiple sinks, one can determine *how* the probability measure exits  $X_J$ .



## Multiple FSP sinks to analyze switch decisions

---

Using the FSP to determine  
initial switch decisions.



# 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.

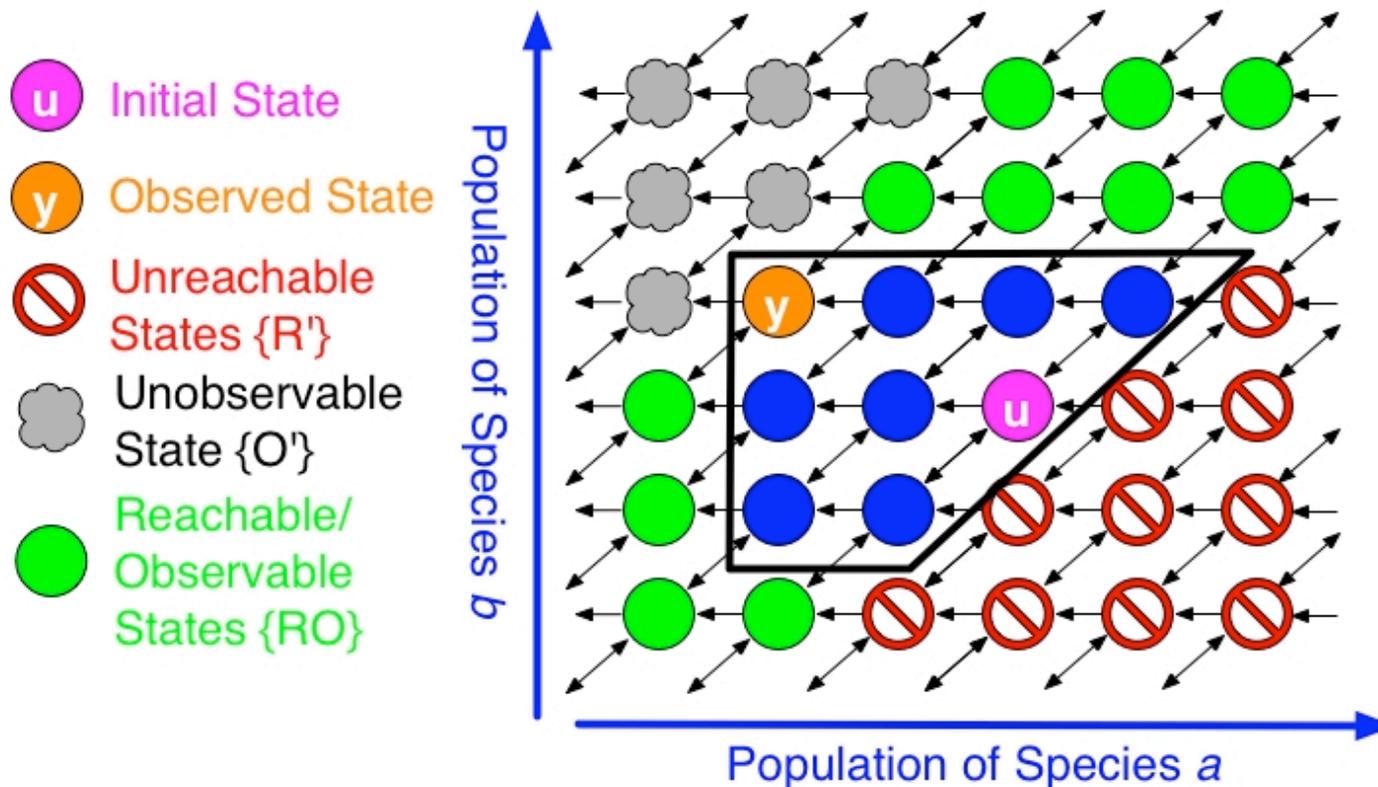
---

- 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  $\mathbf{y}(t)$ :

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

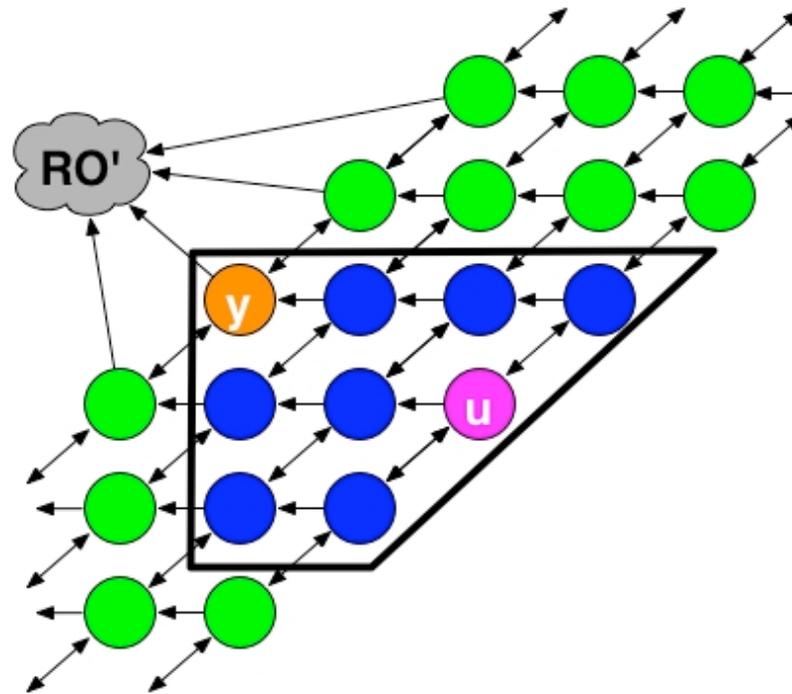
$$\mathbf{y}(t) = \mathbf{C}\mathbf{P}(t)$$

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



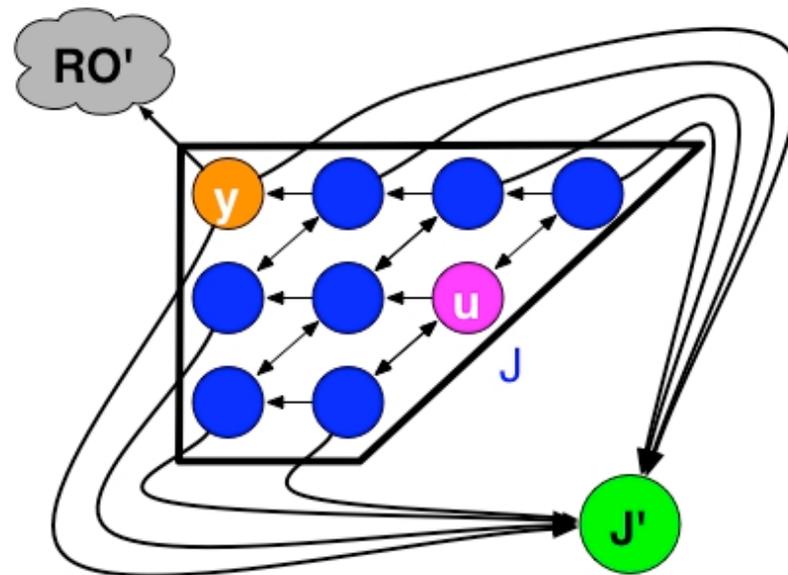
# Remove Unreachable States and Aggregate the Observable States.

---



# 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

---

 Finite State Projection (FSP)

 Reductions to the FSP

★ Aggregating unobservable states

★ Time interval discretization

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

Burrage et al, *A.A. Markov 150th Anniv. Meeting*, 2006

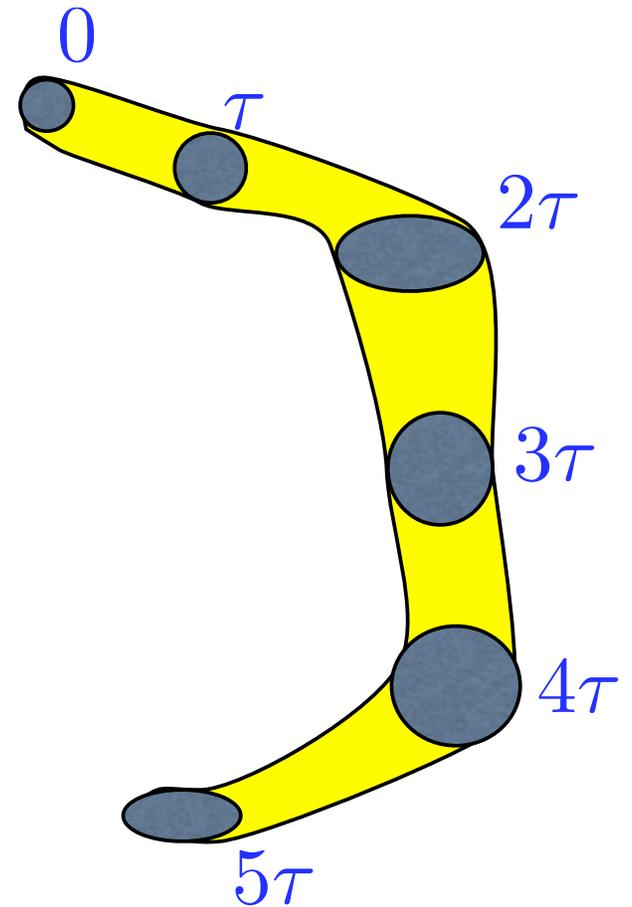
★ Slow manifold projection

★ 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.

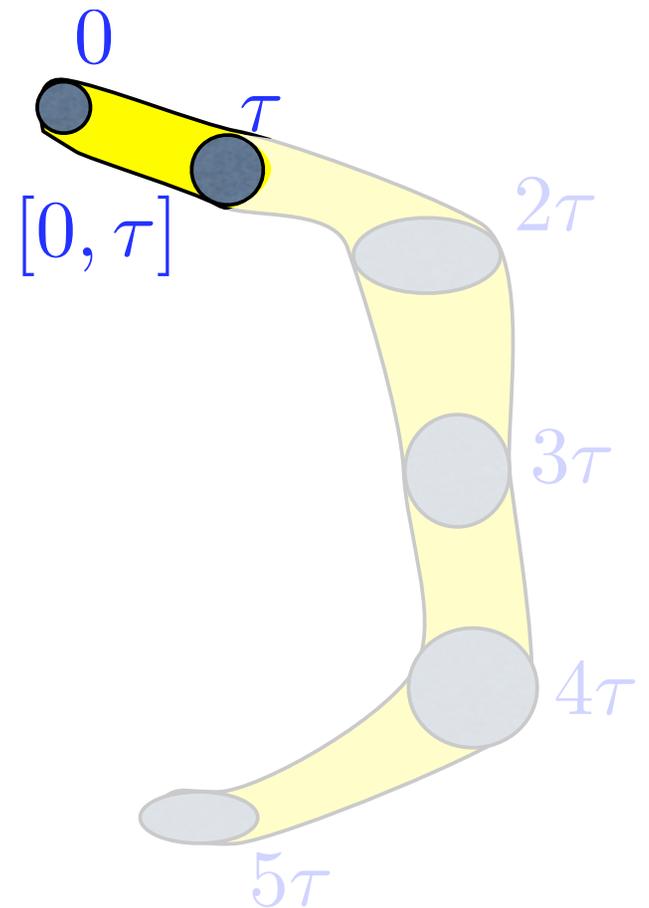


# Time Interval Discretization for the FSP

---

★ Instead:

- \* Discretize the time interval into smaller steps and solve a separate projection for each interval.

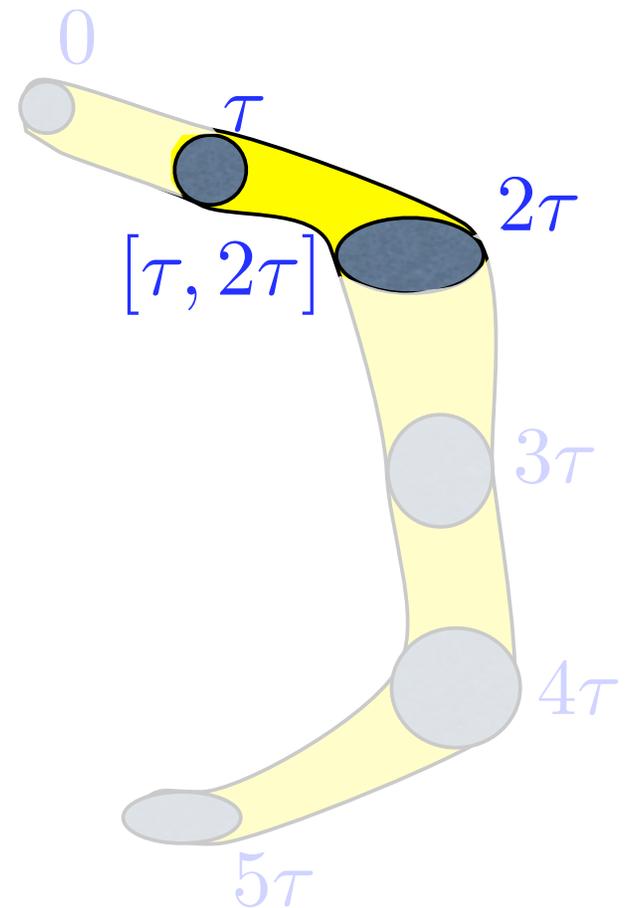


# Time Interval Discretization for the FSP

---

★ Instead:

- \* Discretize the time interval into smaller steps and solve a separate projection for each interval.

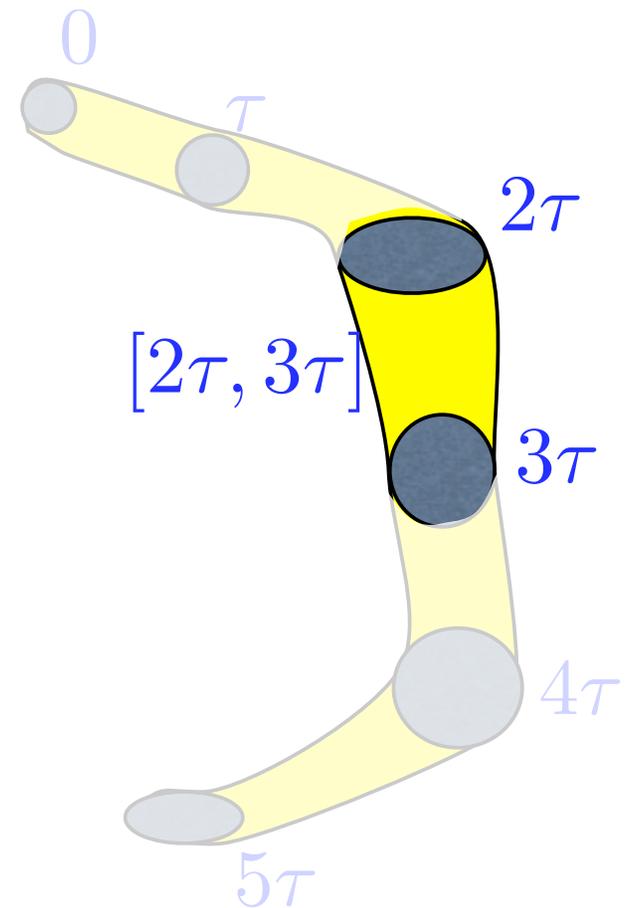


# Time Interval Discretization for the FSP

---

★ Instead:

- \* Discretize the time interval into smaller steps and solve a separate projection for each interval.

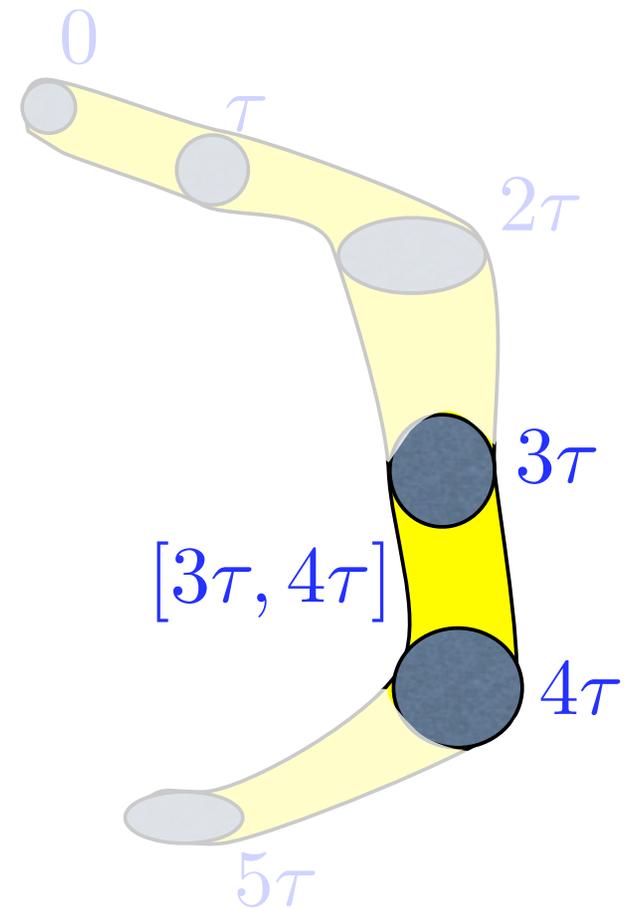


# Time Interval Discretization for the FSP

---

★ Instead:

- \* Discretize the time interval into smaller steps and solve a separate projection for each interval.

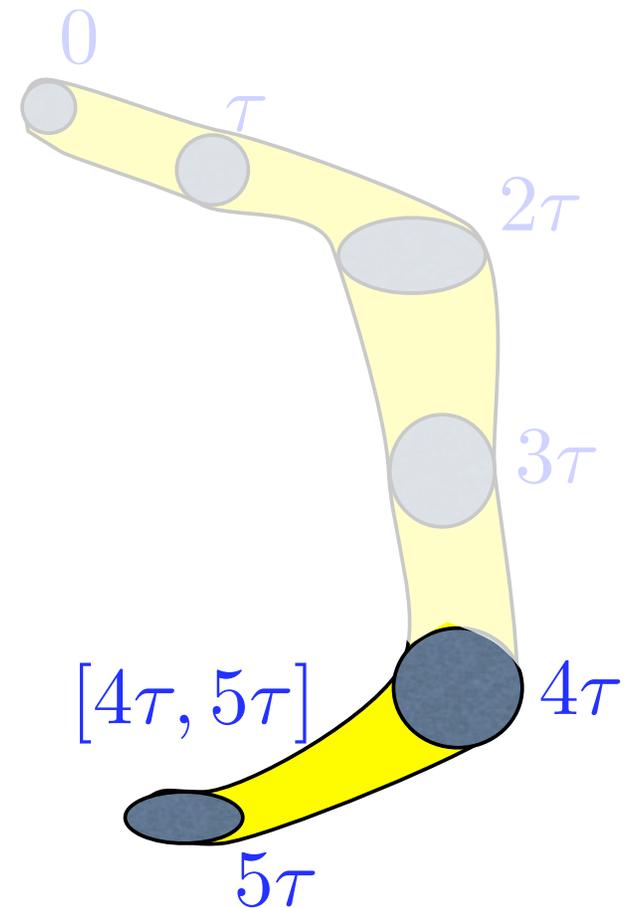


# Time Interval Discretization for the FSP

---

★ Instead:

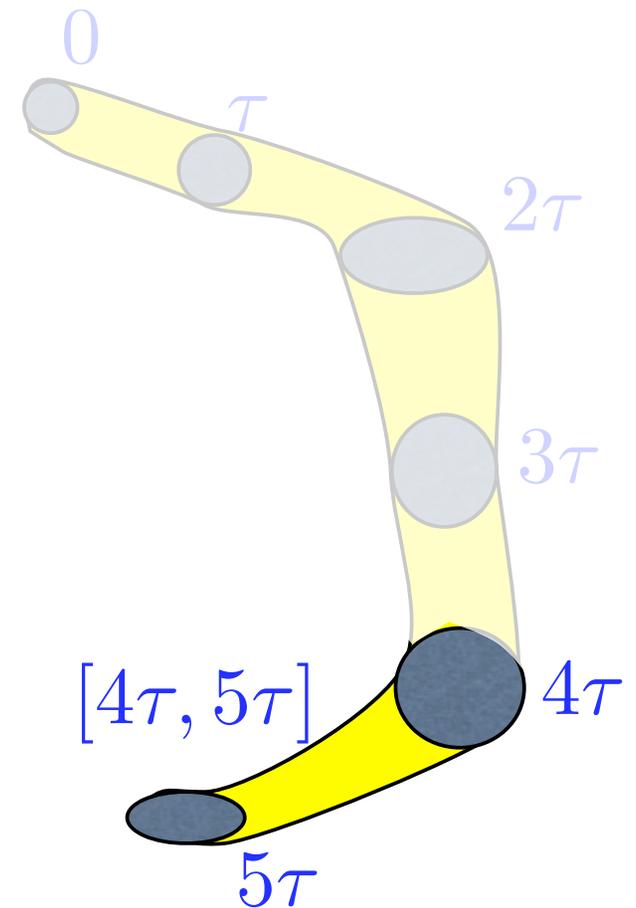
- \* Discretize the time interval into smaller steps and solve a separate projection for each interval.



# 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

---

 Finite State Projection (FSP)

 Reductions to the FSP

- ★ Aggregating unobservable states
- ★ Time interval discretization
- ★ Slow manifold projection  
    Peles/Munsky/Khammash, *JCP*, 2006
- ★ Coarse meshes for the CME.

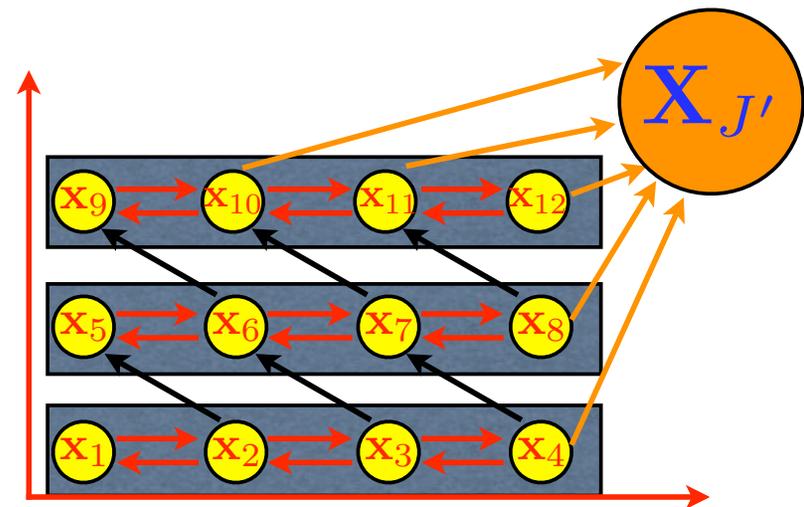
# Perturbation Theory and the FSP

---

- 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 (Slow Manifold Projection)

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



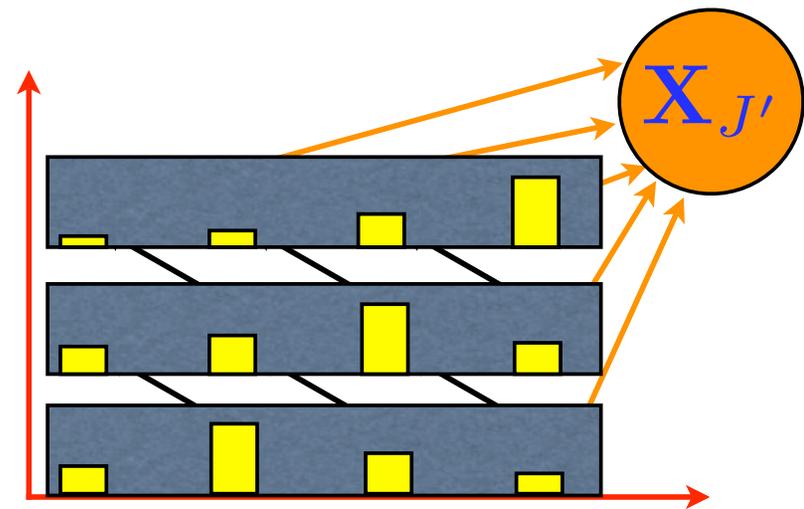
Red Arrows = Fast (Frequent) Reactions

Black Arrows = Slow (Rare) Reactions

Orange Arrows = (Rare) Transitions to Sink

# Intuition (Slow Manifold Projection)

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

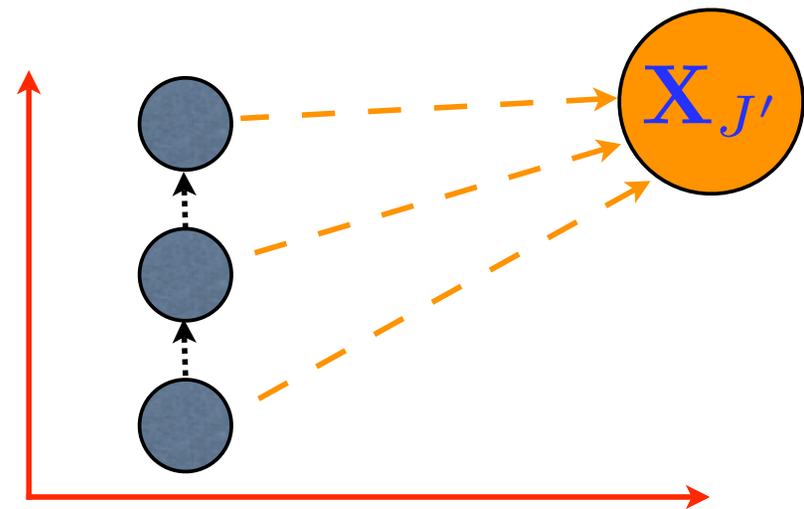
Black Arrows = Slow (Rare) Reactions

Orange Arrows = (Rare) Transitions to Sink

## Intuition (Slow Manifold Projection)

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



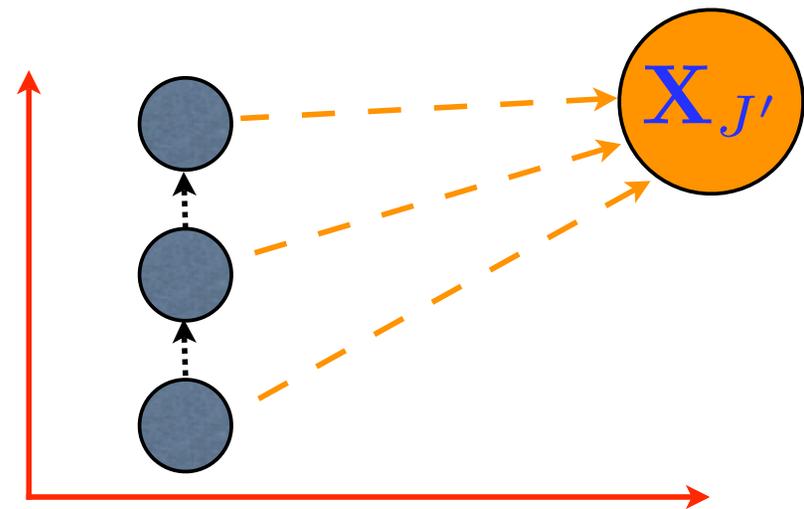
Dotted Black = Averaged Slow Reactions

Dashed Orange = Averaged Transitions to Sink

## Intuition (Slow Manifold Projection)

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

---

## Finite State Projection (FSP)

### Reductions to the FSP

- ★ Aggregating unobservable states
- ★ Time interval discretization
- ★ Slow manifold projection
- ★ Coarse meshes for the CME

Munsky/Khammash, *IEEE Trans. on Auto. Control*, 2008

## Coarse mesh approximation of the CME

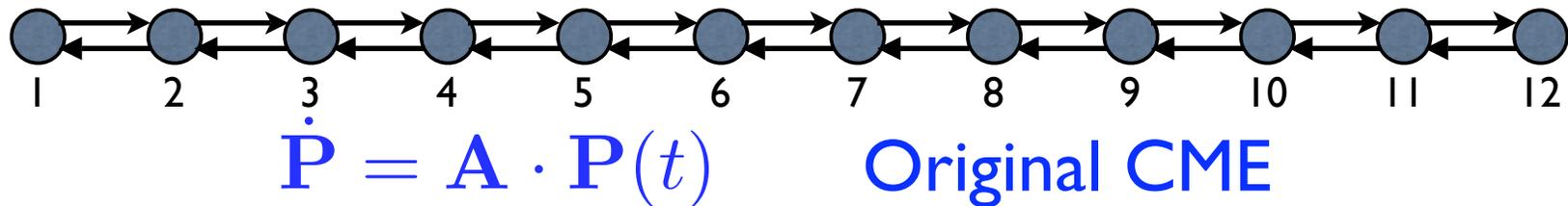
---

- 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.

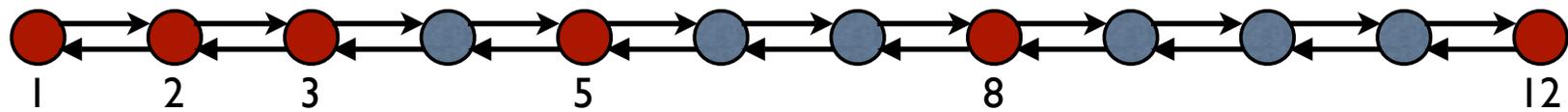
# Coarse mesh approximation of the CME

---

Start with the full 1-dimensional Markov lattice.



Choose a subset of mesh points.



and specify an approximate relation for the

probability of the removed points:  $\mathbf{P} \approx \Phi \mathbf{q}(t)$

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

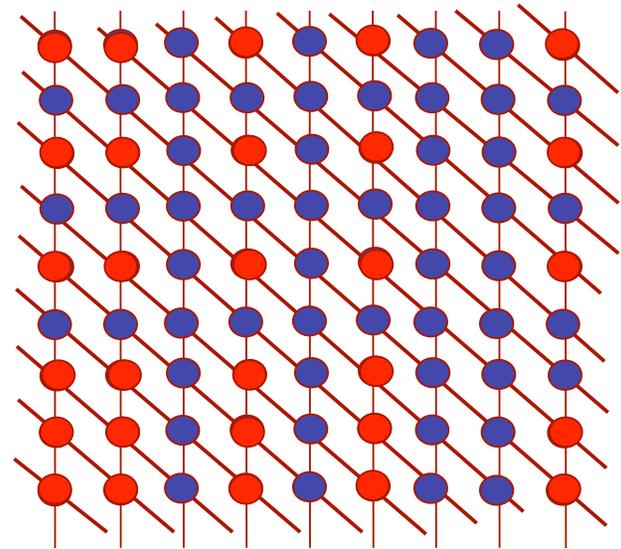
and lift back to the original system coordinates:

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

## Coarse Mesh: Multiple-species problems.

---

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



# Outline

---

Finite State Projection (FSP)

Reductions to the FSP

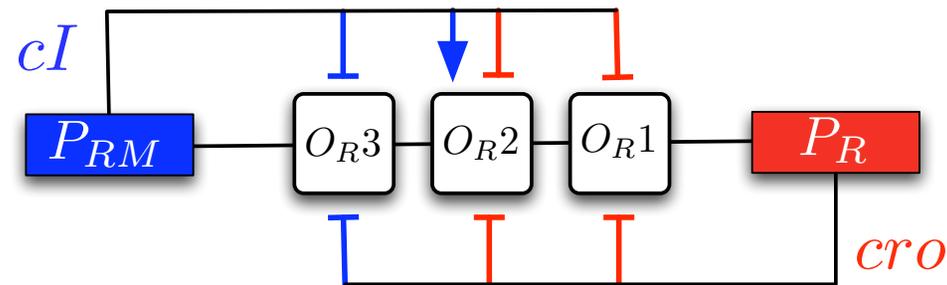
 Case Studies

★ Lambda Phage.

★ Heat Shock.

# A toy model of phage lambda

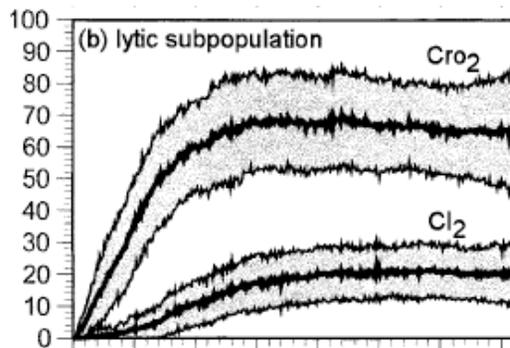
---



- We consider only the core of the lambda switch.
- Two proteins,  $cI$  and  $cro$ .
- These activate and repress the  $P_R$  and  $P_{RM}$  promoters according to the model of Shea and Ackers, 1985.

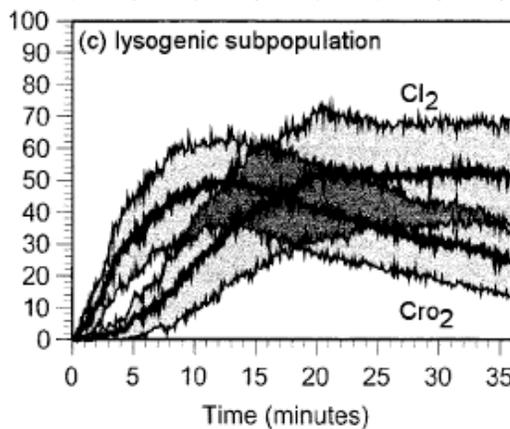
# The Phage Lambda Lysis-Lysogeny Decision

Arkin, Ross, McAdams, 1998.  
Full Model



Lytic  
fate

- ★ Cro reaches a high level before Cl is produced in much quantity.
- ★ Cro represses transcription of Cl.

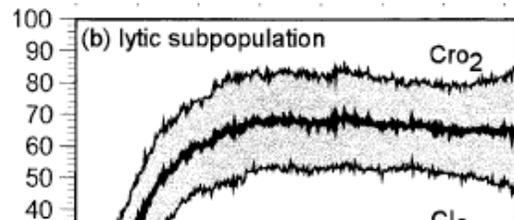


Lysogenic  
fate

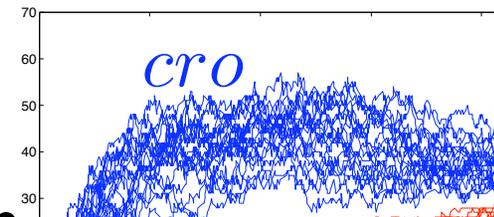
- ★ Cl increases a little earlier.
- ★ Cl represses transcription of Cro.
- ★ Cl is free to increase even further.

# Relevance of Current Model

Arkin, Ross, McAdams, 1998.  
Full Model

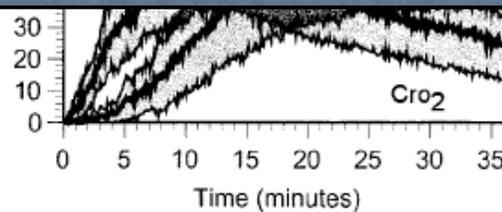


Current simplified model

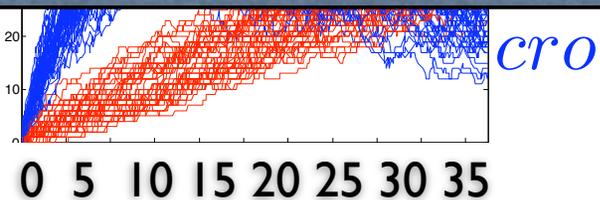


Lytic

Our simplified model captures the important qualitative trends of the cro/cl switch.

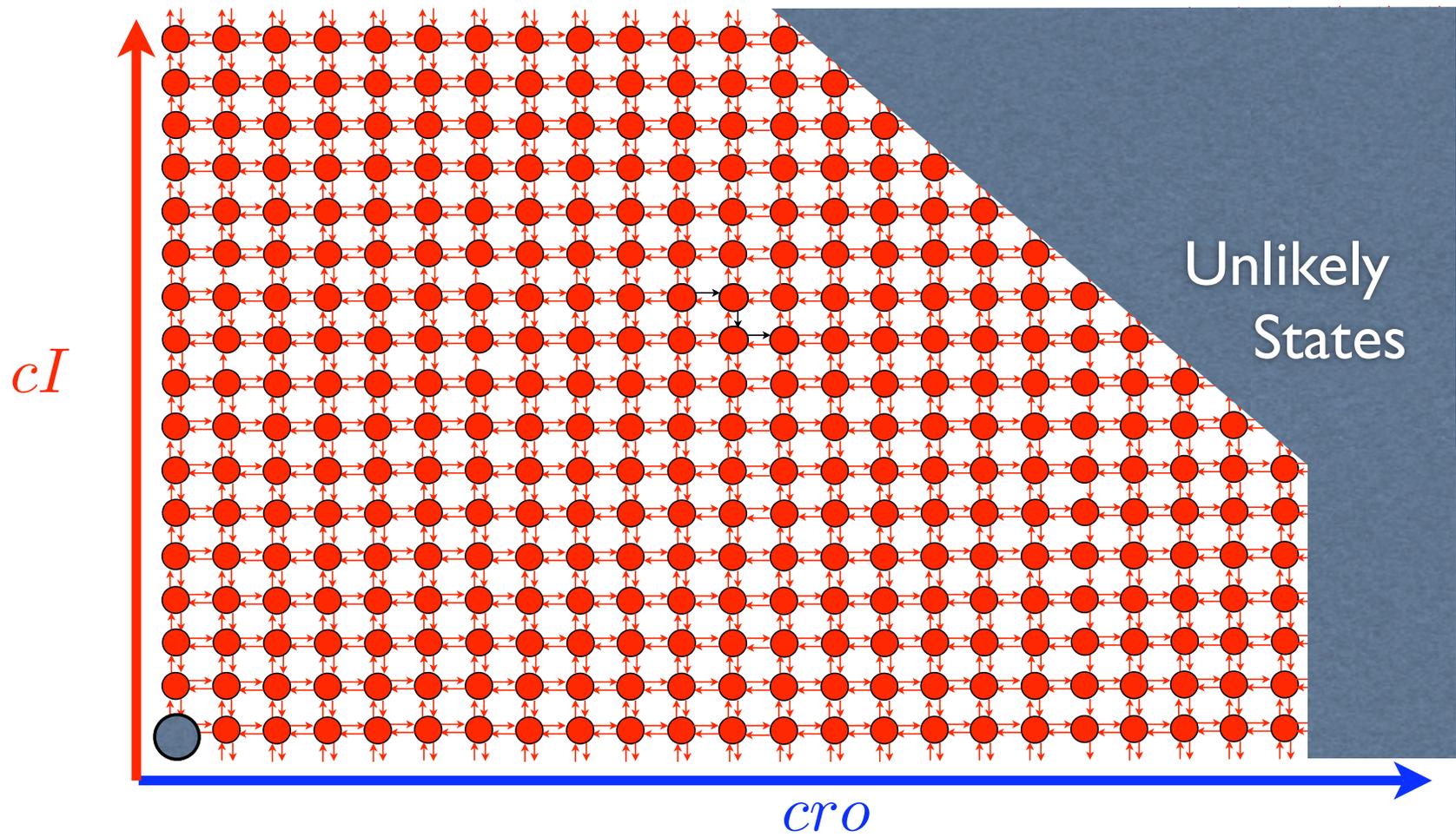


subpopulation

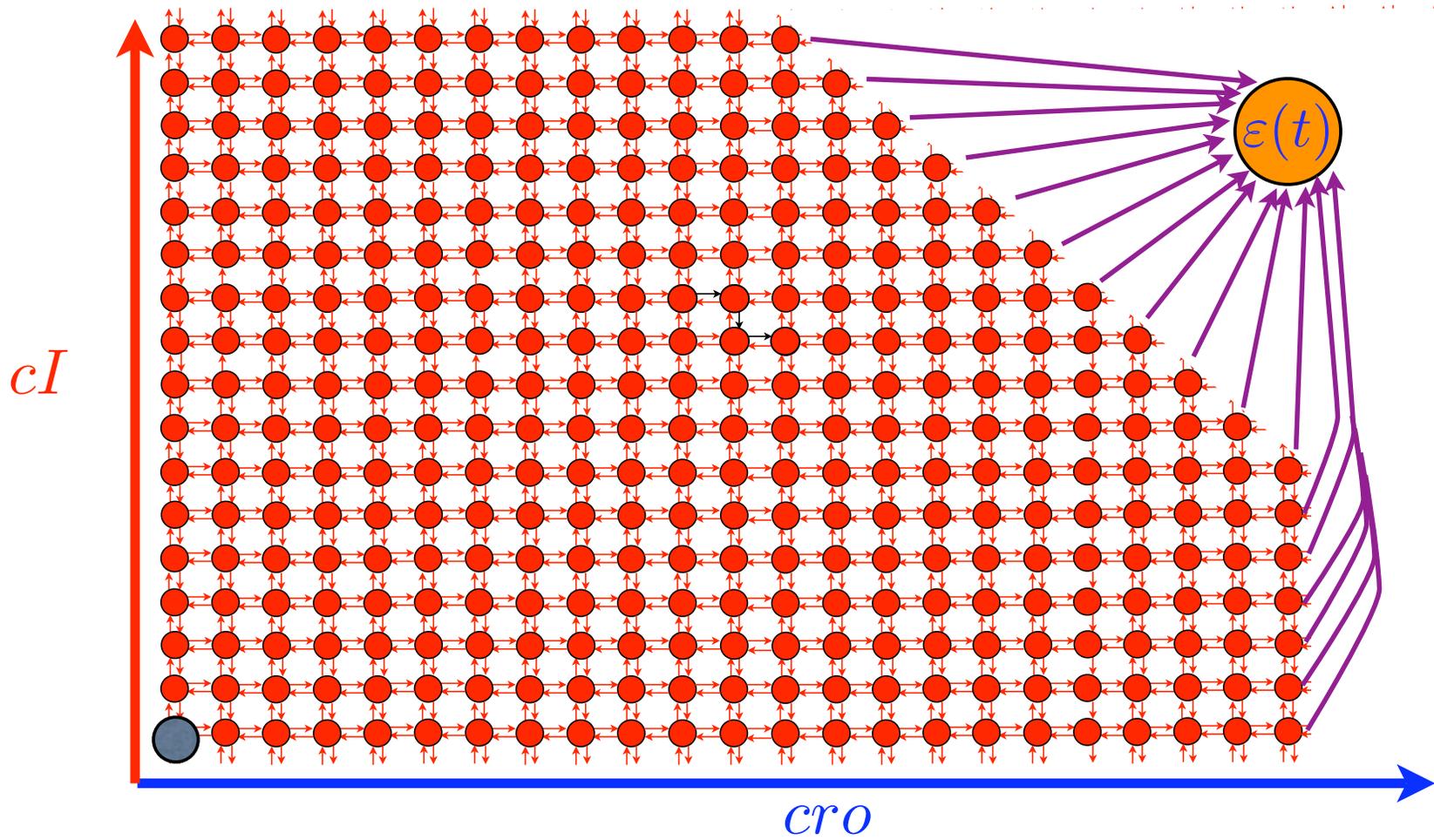


Computations done using Gillespie's SSA.

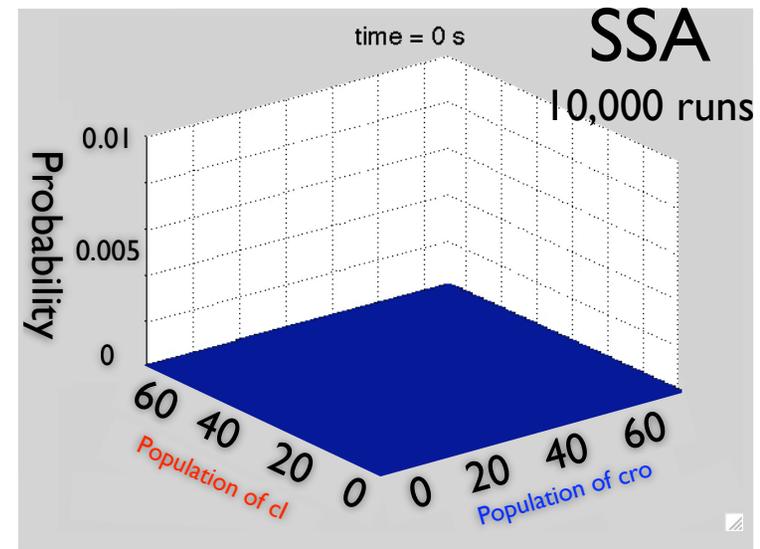
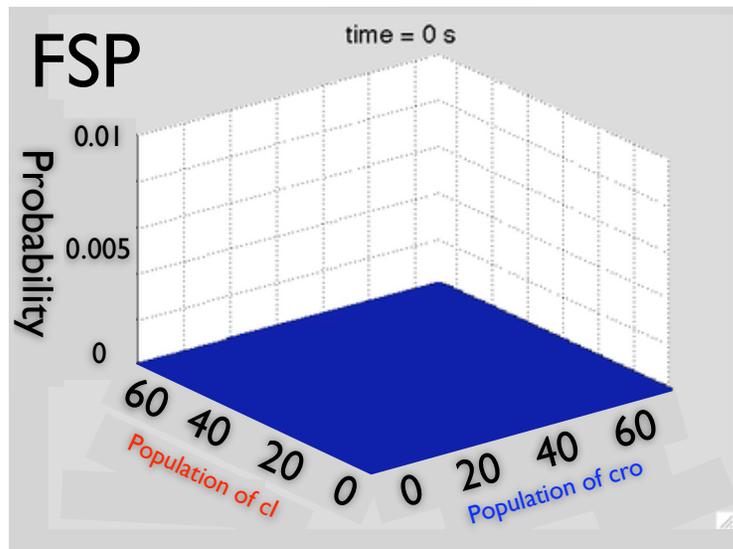
# Applying the FSP to the Phage Lambda Switch



# Applying the FSP to the Phage Lambda Switch



# Efficiency and Accuracy of FSP Results



Method	# Simulations	Time (s)	$\ \text{Error}\ _1$
<b>FSP</b>	$-^a$	163	$\leq 5.3 \times 10^{-3}$

**Guaranteed**  
**No**  
**Guarantees**

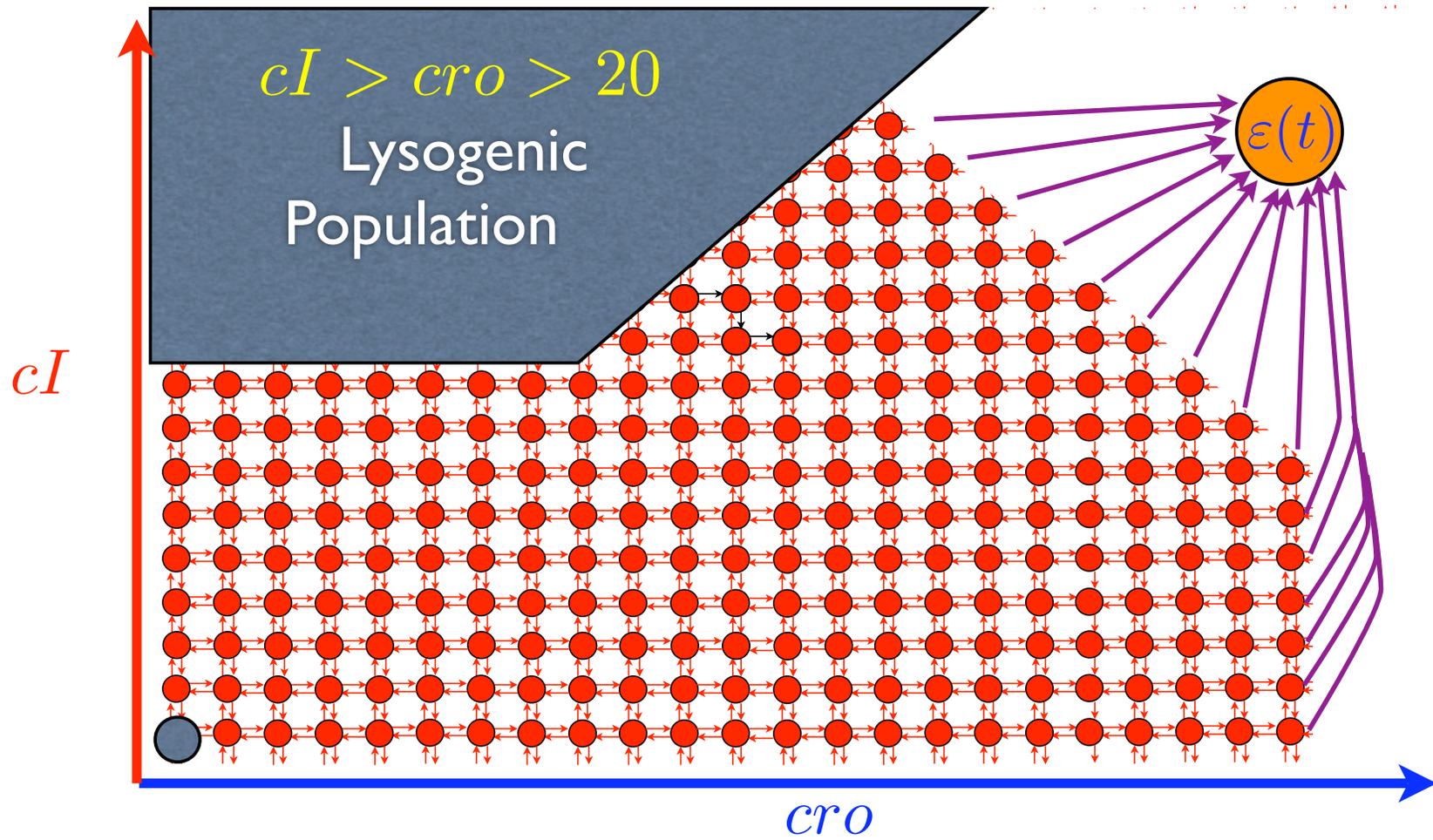
<sup>a</sup>The FSP algorithm is run only once.

## Additional information available with the FSP solution

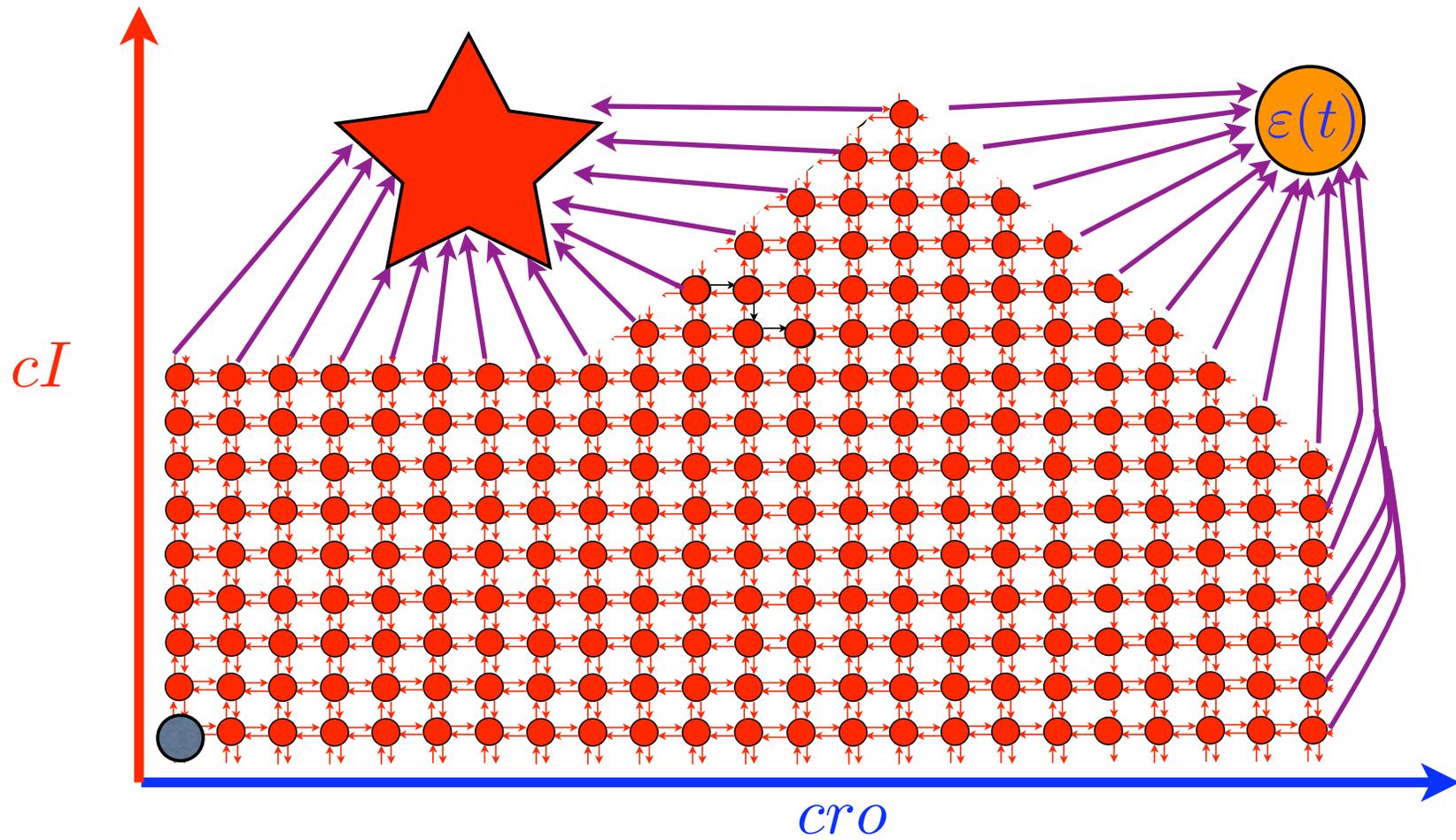
---

- In many cases the FSP is faster and more accurate than Monte Carlo methods.
- Higher precision allows greater flexibility.
  - ★ Direct Computation of Switch Rates.

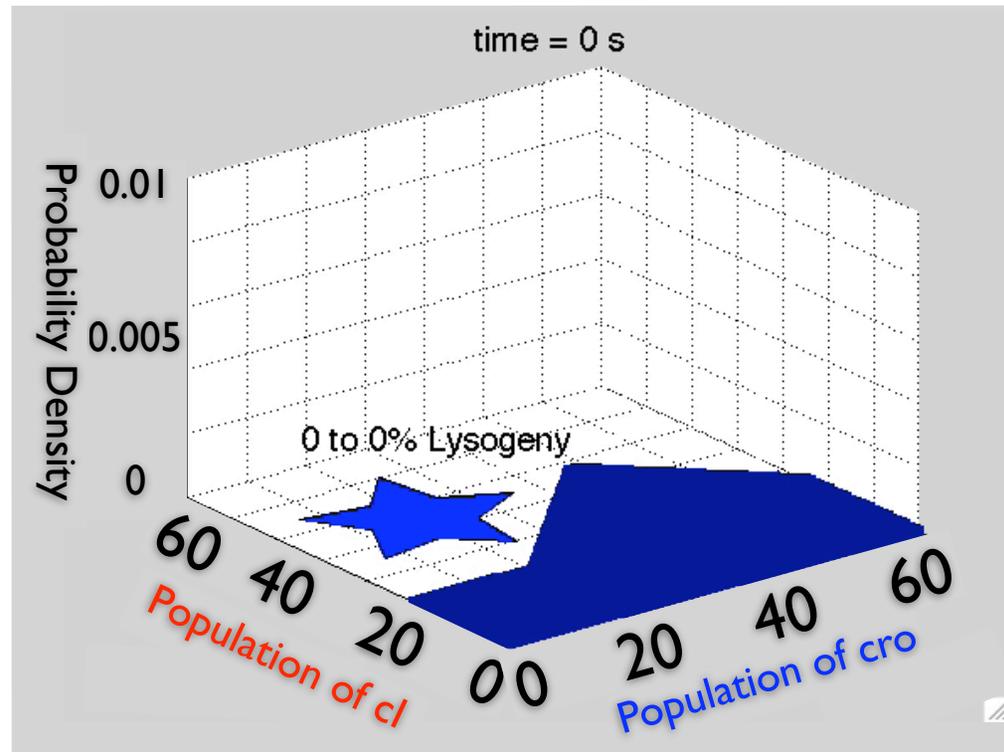
# Using the FSP to Compute Switch Rates



# Using the FSP to Compute Switch Rates



# Using the FSP to Compute Switch Rates



Method	Time (s)	Relative Error	Guarantee?
<b>FSP</b>	25.5 s	< 0.08 %	yes
$10^4$ SSA runs	440.0 s	$\approx 0.90$ %	no

## Additional information available with the FSP solution

---

- In many cases the FSP is faster and more accurate than Monte Carlo methods.
- Higher precision allows greater flexibility.
  - ★ Direct Computation of Switch Rates.
  - ★ Simultaneous consideration of many different initial conditions.

# Comparing different initial conditions.

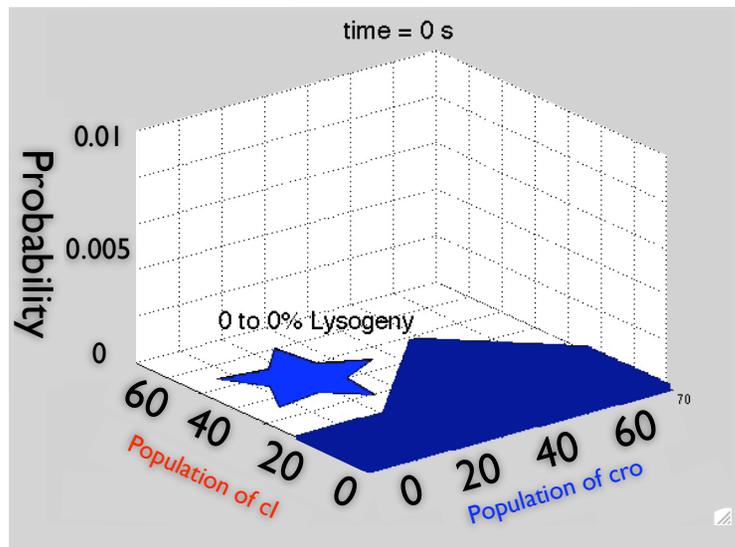
---



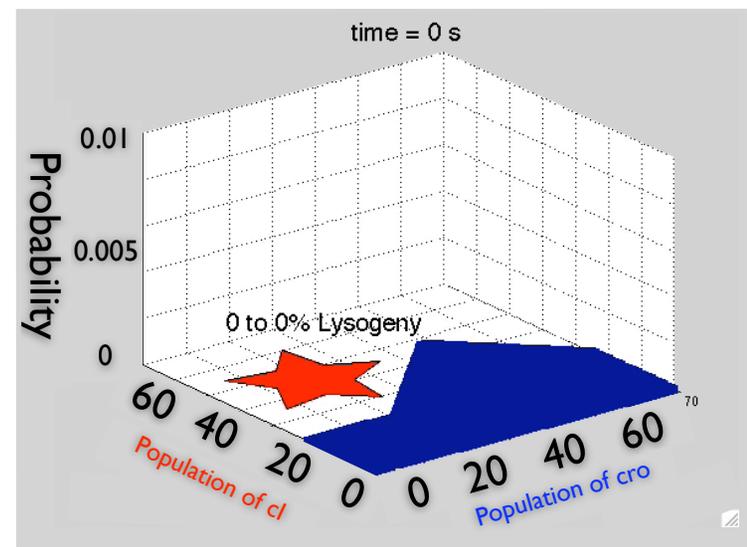
- The FSP is an approximate map of distributions from one time to another.
- This map is valid for any initial distribution.
  - ★ Once computed, this map is cheap to apply again and again.
  - ★ The map automatically provides error bounds for any initial condition!

# Comparing different initial conditions. (Increase in $cro$ )

$$cI_0 = 0$$
$$cro_0 = 0$$



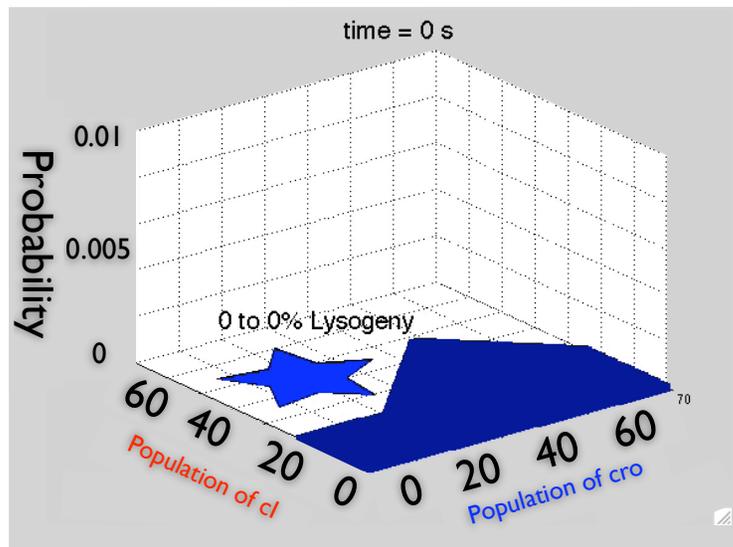
$$cI_0 = 0$$
$$cro_0 = 5$$



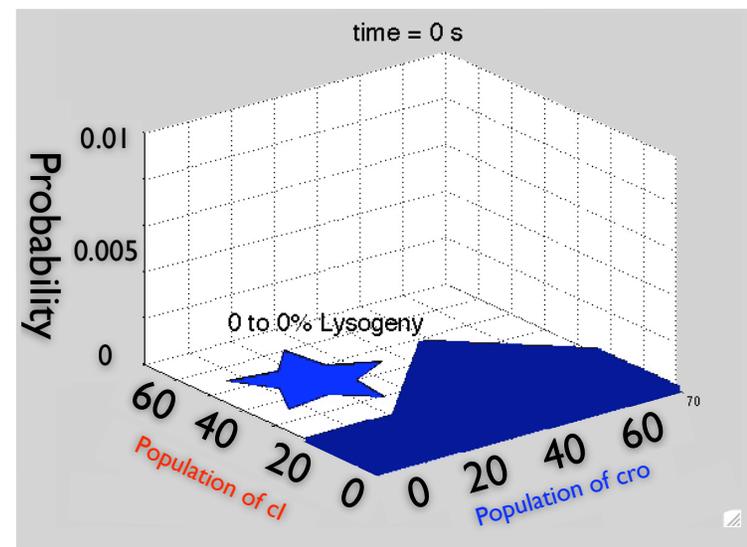
Increasing the initial amount of  $cro$  yields a slight decrease in the lysogeny rate.

# Comparing different initial conditions. (Increase in $cI$ )

$$cI_0 = 0$$
$$cro_0 = 0$$

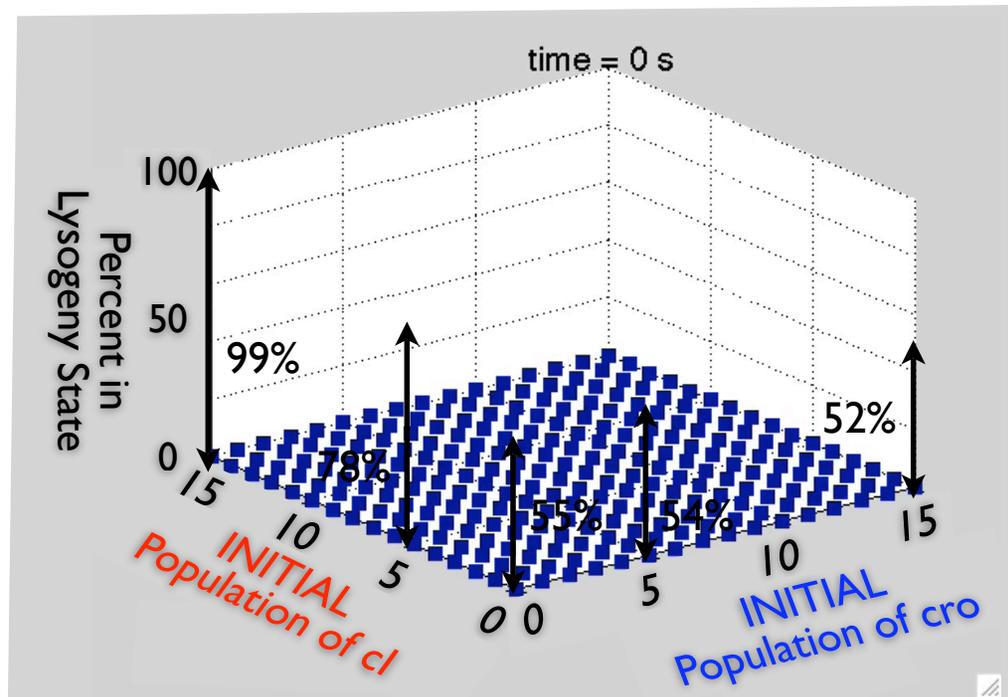


$$cI_0 = 5$$
$$cro_0 = 0$$



Increasing the initial amount of  $cI$  yields a significant increase in lysogeny rate.

# Simultaneous comparison of an array of initial condition.)



Method	Time (s)	# I.C.'s	$\ Error\ _1$	Guarantee?
<b>FSP</b>	66.9 s	2000	$< 1 \times 10^{-4}$	yes
$10^4$ SSA runs	440.0 s	1	$\approx 0.09$	no
$10^{13}$ SSA runs	$\approx 14,000$ years!	2000	$\approx 1 \times 10^{-4}$	no

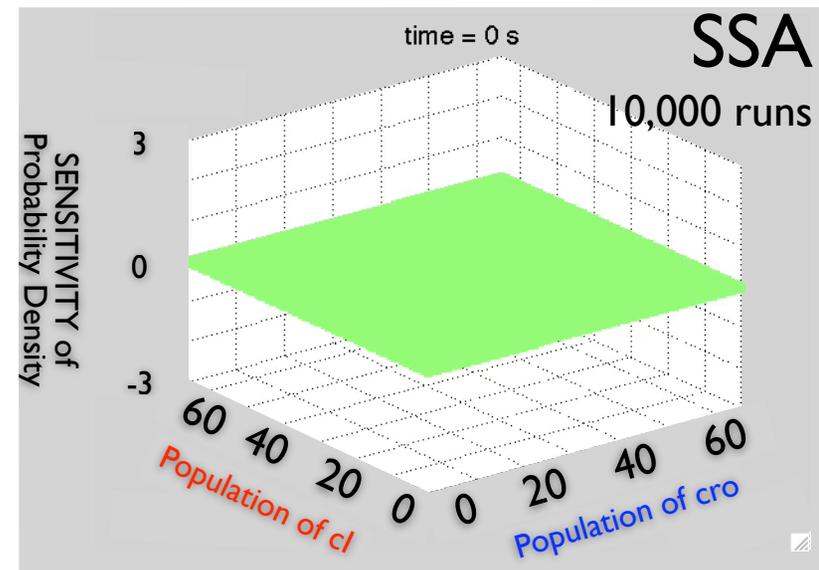
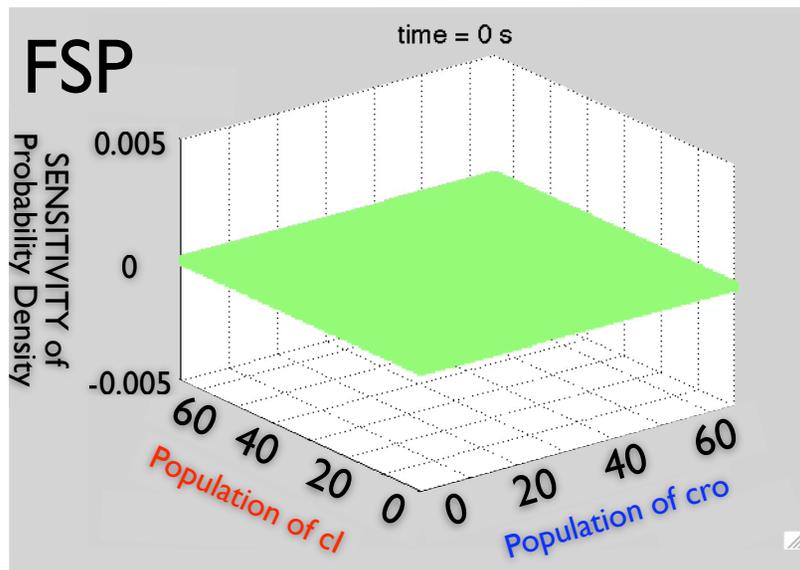
## Additional information available with the FSP solution

---

- In many cases the FSP is both faster and more accurate than other available methods.
- Higher precision allows greater flexibility.
  - ★ Direct Computation of Switch Rates.
  - ★ Simultaneous consideration of many different initial conditions.
  - ★ Sensitivity to parameter changes.

# Parametric Sensitivity of Probability Distributions.

Sensitivity to a small increase in cell Volume.



- ★ Sensitivity analysis requires a huge degree of accuracy.
- ★ Monte Carlo methods would require hundreds of millions of runs!!

# Outline

---

Finite State Projection (FSP)

Reductions to the FSP

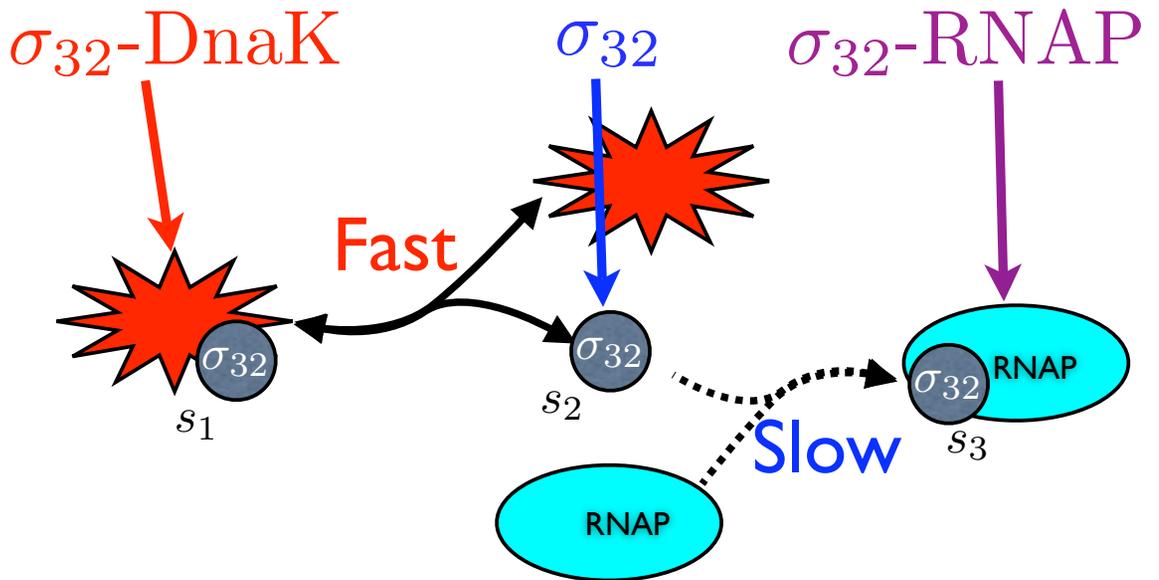
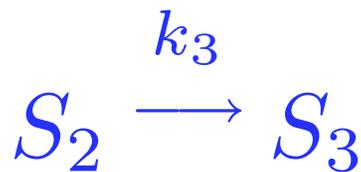
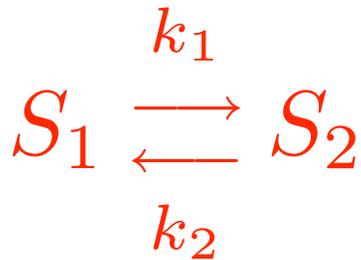
 Case Studies

★ Lambda Phage.

★ Heat Shock.

# Toy Heat Shock Model in *E. coli*

3 forms for  $\sigma_{32}$ :



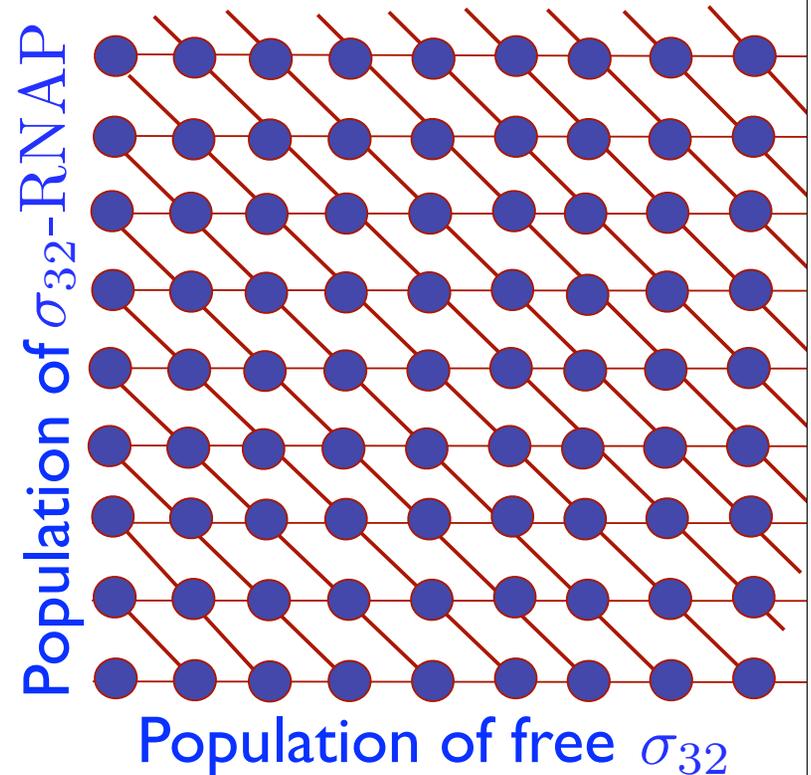
El Samad et al, *PNAS*, vol. 102, No. 8, 2005

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

---

Five Different FSP Solution  
Schemes:

I. Full FSP



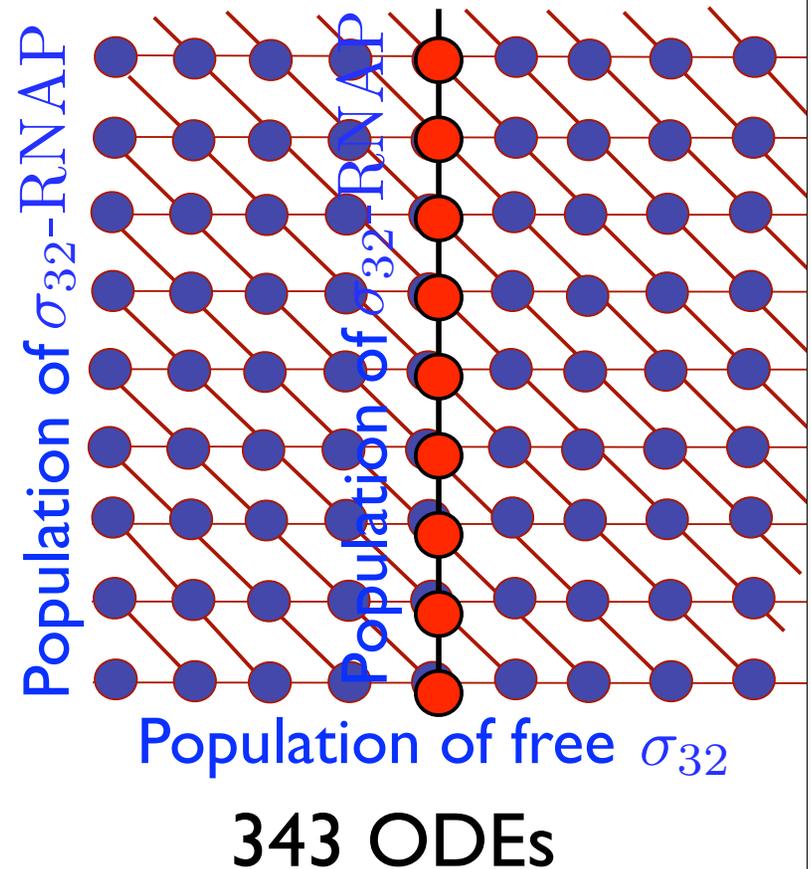
4459 ODEs

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

---

Five Different FSP Solution  
Schemes:

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

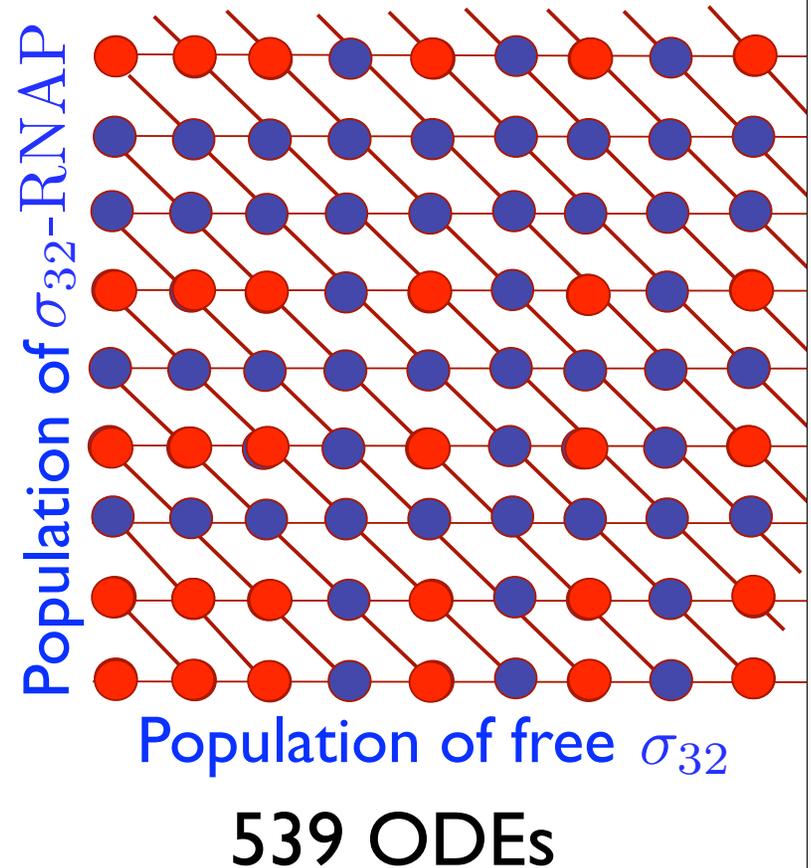


# Toy Heat Shock Model in *E. coli* (cont.)

---

Five Different FSP Solution  
Schemes:

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

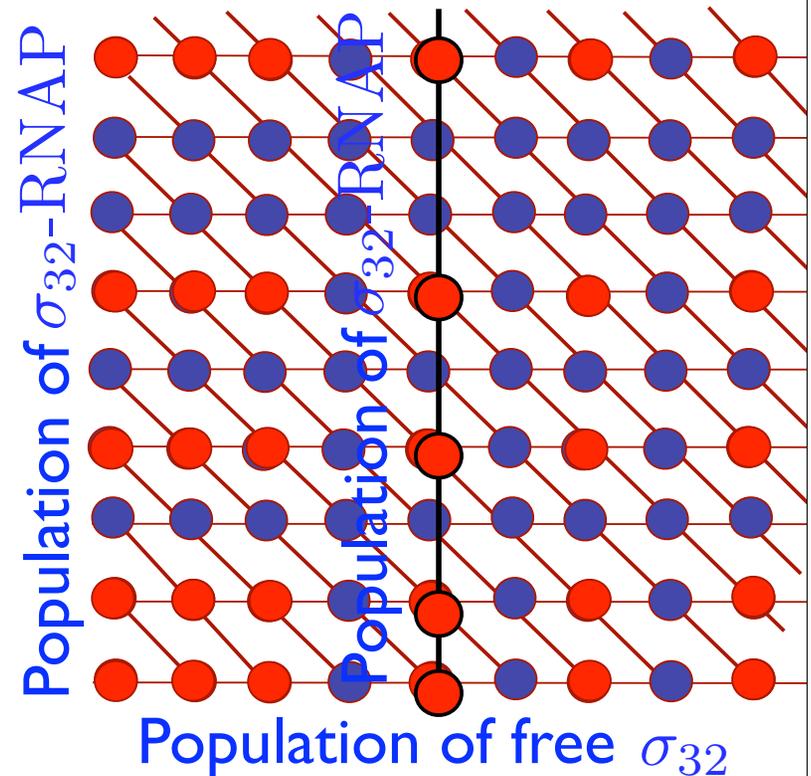


# Toy Heat Shock Model in *E. coli* (cont.)

---

## Five Different FSP Solution Schemes:

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



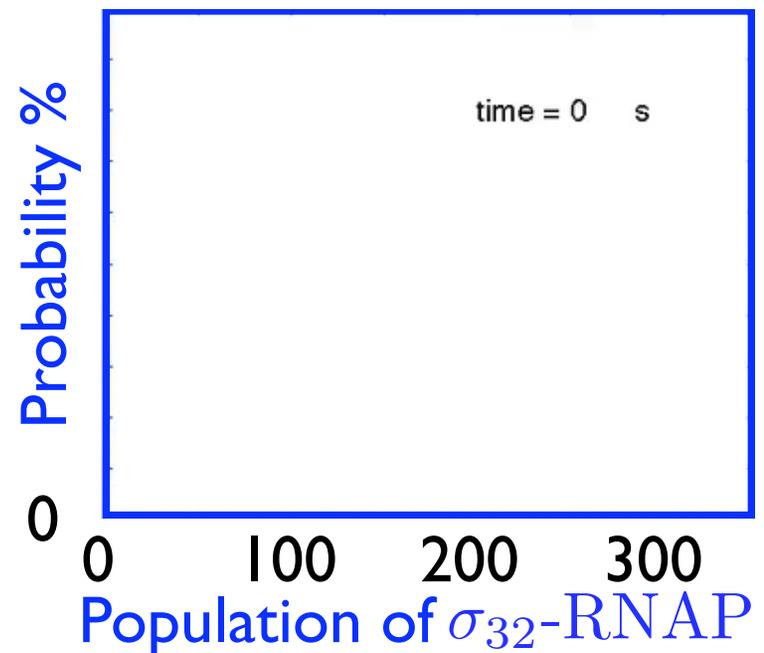
49 ODEs

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

---

## Five Different FSP Solution Schemes:

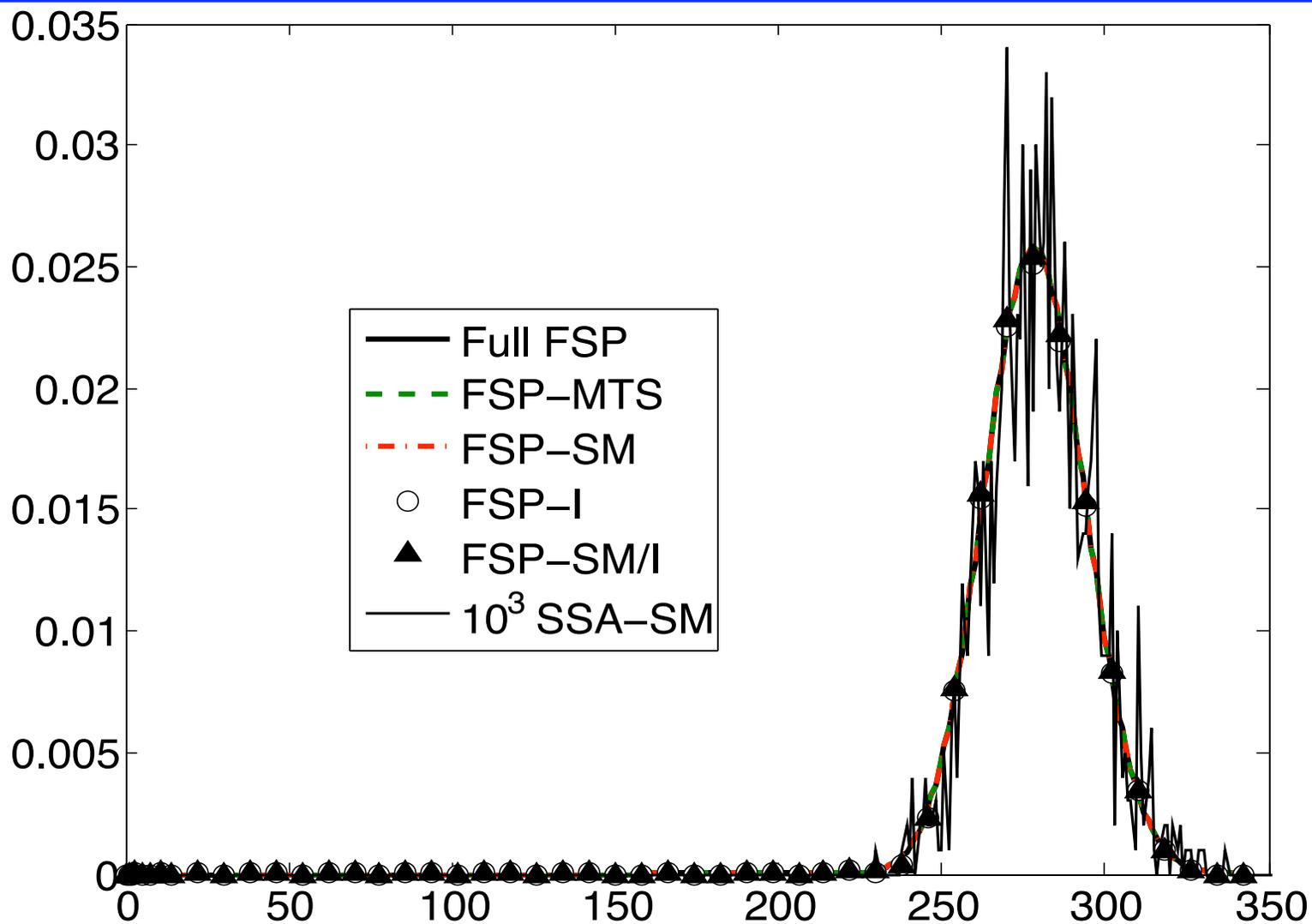
1. Full FSP
2. Slow manifold (FSP-SM)
3. Interpolated (FSP-I)
4. Hybrid (FSP-SM/I)
5. Multiple time interval  
(FSP-MTI)



70 sets of 195 or fewer ODEs.

## Efficiency and accuracy of the reduced FSP methods

---



## Efficiency and accuracy of the reduced FSP methods

For final time $t_f = 300s$				
Method	Matrix Size	$J_{solve}$	$J_{total}$	$\infty$ -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	$\approx 0.0116$
$10^4$ 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!

# 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  $[0, t]$ )