Introduction to stochastic modelling in Mathematical Biology

Tomás Alarcón

Computational & Mathematical Biology Group
Centre de Recerca Matemàtica

Outline

Numerical methods I: Gillespie stochastic simulation algorithm

Numerical Methods II: The τ-leap method

Outline

Numerical methods I: Gillespie stochastic simulation algorithm

Numerical Methods II: The τ-leap method

Gillespie stochastic simulation algorithm[1]

- Gillespie's algorithm is a Monte Carlo simulation method which generates sample paths or realisations of a Markov processes. The statistical properties of the ensemble of such sample paths converge, when the number of realisations tends to infinity, to the solution of the corresponding Master Equation

[1] D.T. Gillespie. J. Comp. Phys. **22**, 403 (1976)

Gillespie stochastic simulation algorithm[1]

- Gillespie's algorithm is a Monte Carlo simulation method which generates sample paths or realisations of a Markov processes. The statistical properties of the ensemble of such sample paths converge, when the number of realisations tends to infinity, to the solution of the corresponding Master Equation
- The algorithm is based on an exact representation of the Master Equation. In this sense, the sample paths generated using the Gillespie algorithm are *exact* realisations of the underlying Markov process

[1] D.T. Gillespie. J. Comp. Phys. **22**, 403 (1976)

Gillespie stochastic simulation algorithm[1]

- Gillespie's algorithm is a Monte Carlo simulation method which generates sample paths or realisations of a Markov processes. The statistical properties of the ensemble of such sample paths converge, when the number of realisations tends to infinity, to the solution of the corresponding Master Equation
- The algorithm is based on an exact representation of the Master Equation. In this sense, the sample paths generated using the Gillespie algorithm are *exact* realisations of the underlying Markov process
- The mathematical foundation of the numerical algorithm is based on a (rather clever) reinterpretation of the Master Equation, with the so-called elementary process probability, $P(\tau, i)$, as the central element

[1] D.T. Gillespie. J. Comp. Phys. **22**, 403 (1976)

Derivation of the algorithm I

- Recall that $W_i(X(t))\Delta t$ is the probability of event i to occur between t and $t + \Delta t$

Derivation of the algorithm I

- Recall that $W_i(X(t))\Delta t$ is the probability of event i to occur between t and $t + \Delta t$
- The elementary process probability is such that $P(\tau, i)\Delta t$ is the probability of the next process to occur in $(t + \tau, t + \tau + \Delta t)$ and be process i

Derivation of the algorithm I

- Recall that $W_i(X(t))\Delta t$ is the probability of event i to occur between t and $t+\Delta t$
- The elementary process probability is such that $P(\tau, i)\Delta t$ is the probability of the next process to occur in $(t+\tau, t+\tau+\Delta t)$ and be process i
- $P(\tau, i)\Delta t$ can be written as the product of the probability of no event occurring between $(t, t+\tau)$, $P_0(\tau)$, and the probability that event i occurs between $t+\tau$ and $t+\tau+\Delta t)$, i.e. $W_i(X(t))\Delta t$:

$$P(\tau, i)\Delta t = P_0(\tau) W_i(X(t))\Delta t$$

Derivation of the algorithm I

- Recall that $W_i(X(t))\Delta t$ is the probability of event i to occur between t and $t+\Delta t$
- The elementary process probability is such that $P(\tau, i)\Delta t$ is the probability of the next process to occur in $(t+\tau, t+\tau+\Delta t)$ and be process i
- $P(\tau, i)\Delta t$ can be written as the product of the probability of no event occurring between $(t, t+\tau)$, $P_0(\tau)$, and the probability that event i occurs between $t+\tau$ and $t+\tau+\Delta t)$, i.e. $W_i(X(t))\Delta t$:

$$P(\tau, i)\Delta t = P_0(\tau)W_i(X(t))\Delta t$$

- The next step in our derivation is calculating $P_0(\tau)$

Derivation of the algorithm II

Derivation of $P_0(\tau)$

1. Divide $(t, t+\tau)$ into $k \gg 1$ equal intervals of duration $\epsilon = \tau/k$

Derivation of the algorithm II

Derivation of $P_0(\tau)$

1. Divide $(t, t+\tau)$ into $k \gg 1$ equal intervals of duration $\epsilon = \tau/k$
2. The probability of no reaction occuring during any of these subintervals is given by:

$$\prod_{i=1}^{R} \left(1 - W_i(X(t))\epsilon + O(\epsilon^2)\right) \simeq 1 - \sum_{i=1}^{R} W_i(X(t))\epsilon + O(\epsilon^2)$$

Derivation of the algorithm II

Derivation of $P_0(\tau)$

1. Divide $(t, t+\tau)$ into $k \gg 1$ equal intervals of duration $\epsilon = \tau/k$
2. The probability of no reaction occuring during any of these subintervals is given by:

$$\prod_{i=1}^{R} \left(1 - W_i(X(t))\epsilon + O(\epsilon^2)\right) \simeq 1 - \sum_{i=1}^{R} W_i(X(t))\epsilon + O(\epsilon^2)$$

3. Using the Markov Property and recalling that $\epsilon = \tau/k$, $P_0(\tau)$ can be written as:

$$P_0(\tau) = \left(1 - \sum_{i=1}^{R} W_i(X(t))\frac{\tau}{k} + O(k^{-2})\right)^k$$

Derivation of the algorithm II

Derivation of $P_0(\tau)$

1. Divide $(t, t+\tau)$ into $k \gg 1$ equal intervals of duration $\epsilon = \tau/k$
2. The probability of no reaction occuring during any of these subintervals is given by:

$$\prod_{i=1}^{R} \left(1 - W_i(X(t))\epsilon + O(\epsilon^2)\right) \simeq 1 - \sum_{i=1}^{R} W_i(X(t))\epsilon + O(\epsilon^2)$$

3. Using the Markov Property and recalling that $\epsilon = \tau/k$, $P_0(\tau)$ can be written as:

$$P_0(\tau) = \left(1 - \sum_{i=1}^{R} W_i(X(t))\frac{\tau}{k} + O(k^{-2})\right)^k$$

4. Which, in the limit $k \to \infty$ is $P_0(\tau) = e^{-\tau \sum_{i=1}^{R} W_i(X(t))}$

Derivation of the algorithm III

Finally ...

1. So, the quantity $P(\tau, i)$, i.e. the probability density of the next process to occur in $(t+\tau, t+\tau+\Delta t)$ and be process i, is given by:

$$P(\tau, i) = W_i(X(t))e^{-\tau \sum_{i=1}^{R} W_i(X(t))},$$

Derivation of the algorithm III

Finally ...

1. So, the quantity $P(\tau, i)$, i.e. the probability density of the next process to occur in $(t+\tau, t+\tau+\Delta t)$ and be process i, is given by:

$$P(\tau, i) = W_i(X(t))e^{-\tau \sum_{i=1}^{R} W_i(X(t))},$$

2. Which can be rewitten as $P(\tau, i) = P(\tau|X(t))P(i|\tau, X(t))$ where:

Derivation of the algorithm III

Finally ...

1. So, the quantity $P(\tau, i)$, i.e. the probability density of the next process to occur in $(t+\tau, t+\tau+\Delta t)$ and be process i, is given by:

$$P(\tau, i) = W_i(X(t))e^{-\tau \sum_{i=1}^{R} W_i(X(t))},$$

2. Which can be rewitten as $P(\tau, i) = P(\tau|X(t))P(i|\tau, X(t))$ where:
   - $P(\tau|X(t)) = (\sum_i W_i(X(t))e^{-\tau \sum_i W_i(X(t))}$ is the waiting time distribution conditioned to the state of the system at time t be $X(t)$
   - $P(i|\tau, X(t)) = \frac{W_i(X(t))}{\sum_i W_i(X(t))}$ is the probability of process i to occur conditioned to the waiting time be τ and the state of the system at time t be $X(t)$

Gillespie algorithm

Gillespie stochastic simulation algorithm

1. Initialisation $X(t=0) = X_0$. Initialise seed of random number generator

Gillespie algorithm

Gillespie stochastic simulation algorithm

1. Initialisation $X(t=0) = X_0$. Initialise seed of random number generator
2. Calculate $W_i(X(t))$ for $i = 1, \ldots, R$. Calculate $W_0(X(t)) = \sum_{i=1}^{R} W_i(X(t))$

Gillespie algorithm

Gillespie stochastic simulation algorithm

1. Initialisation $X(t=0) = X_0$. Initialise seed of random number generator
2. Calculate $W_i(X(t))$ for $i = 1, \ldots, R$. Calculate $W_0(X(t)) = \sum_{i=1}^{R} W_i(X(t))$
3. Generate two random numbers z_1 and z_2 uniformly distributed on the unit interval

Gillespie algorithm

Gillespie stochastic simulation algorithm

1. Initialisation $X(t=0) = X_0$. Initialise seed of random number generator
2. Calculate $W_i(X(t))$ for $i = 1, \ldots, R$. Calculate $W_0(X(t)) = \sum_{i=1}^{R} W_i(X(t))$
3. Generate two random numbers z_1 and z_2 uniformly distributed on the unit interval
4. Calculate the waiting time $\tau = \frac{1}{W_0(X(t))} \log\left(\frac{1}{z_1}\right)$

Gillespie algorithm

Gillespie stochastic simulation algorithm

1. Initialisation $X(t=0) = X_0$. Initialise seed of random number generator
2. Calculate $W_i(X(t))$ for $i = 1, \ldots, R$. Calculate $W_0(X(t)) = \sum_{i=1}^{R} W_i(X(t))$
3. Generate two random numbers z_1 and z_2 uniformly distributed on the unit interval
4. Calculate the waiting time $\tau = \frac{1}{W_0(X(t))} \log\left(\frac{1}{z_1}\right)$
5. Calculate which process occurs by choosing j so that

$$\sum_{i=1}^{j-1} W_i(X(t)) \leq z_2 W_0 < \sum_{i=1}^{j} W_i(X(t))$$

Gillespie algorithm

Gillespie stochastic simulation algorithm

1. Initialisation $X(t=0) = X_0$. Initialise seed of random number generator
2. Calculate $W_i(X(t))$ for $i = 1, \ldots, R$. Calculate $W_0(X(t)) = \sum_{i=1}^{R} W_i(X(t))$
3. Generate two random numbers z_1 and z_2 uniformly distributed on the unit interval
4. Calculate the waiting time $\tau = \frac{1}{W_0(X(t))} \log\left(\frac{1}{z_1}\right)$
5. Calculate which process occurs by choosing j so that

$$\sum_{i=1}^{j-1} W_i(X(t)) \leq z_2 W_0 < \sum_{i=1}^{j} W_i(X(t))$$

6. Update $t \leftarrow t + \tau$ and $X(t+\tau) \leftarrow X(t) + r_j$

Gillespie algorithm

Gillespie stochastic simulation algorithm

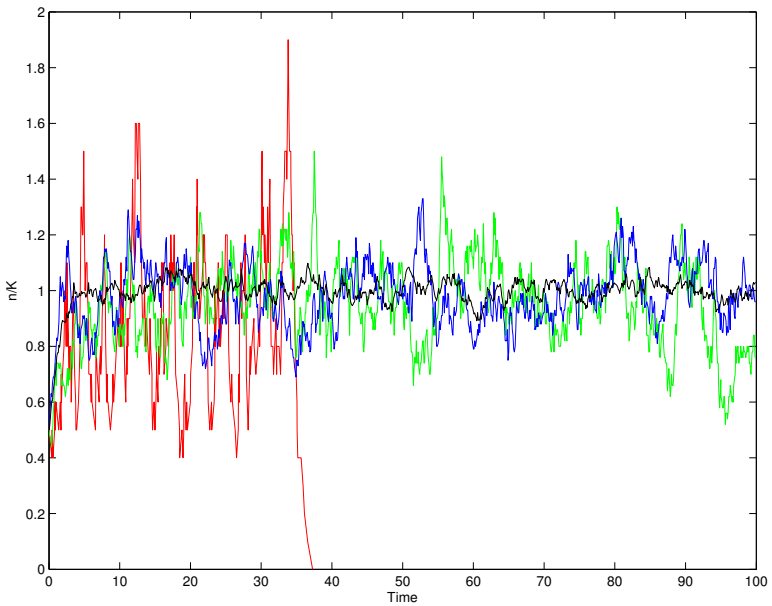
1. Initialisation $X(t=0) = X_0$. Initialise seed of random number generator
2. Calculate $W_i(X(t))$ for $i = 1, \ldots, R$. Calculate $W_0(X(t)) = \sum_{i=1}^{R} W_i(X(t))$
3. Generate two random numbers z_1 and z_2 uniformly distributed on the unit interval
4. Calculate the waiting time $\tau = \frac{1}{W_0(X(t))} \log\left(\frac{1}{z_1}\right)$
5. Calculate which process occurs by choosing j so that

$$\sum_{i=1}^{j-1} W_i(X(t)) \leq z_2 W_0 < \sum_{i=1}^{j} W_i(X(t))$$

6. Update $t \leftarrow t + \tau$ and $X(t+\tau) \leftarrow X(t) + r_j$
7. Iterate steps 2–6 until some stopping criterion is fulfilled (e.g. $t \geq T$)

Example: Branching with binary anihilation[2]

Gillespie SSA sample paths for different values of the carrying capacity $n_s = \sigma/\lambda$

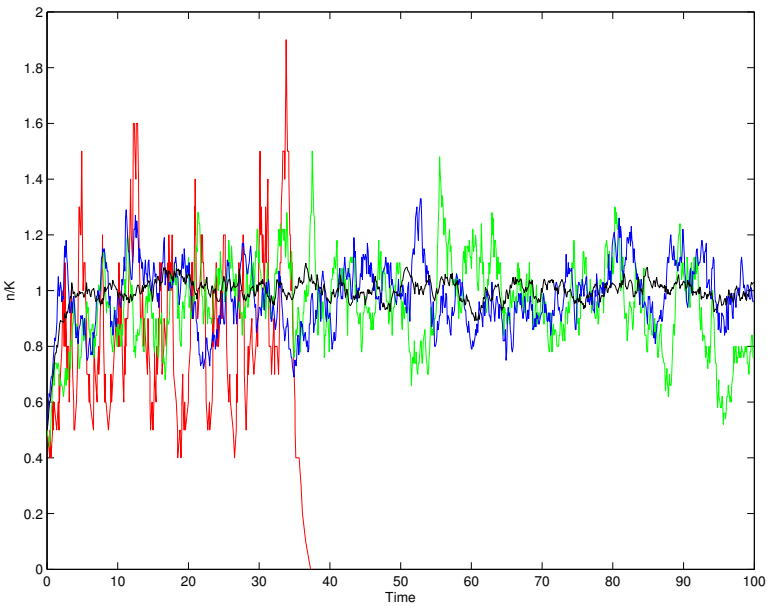


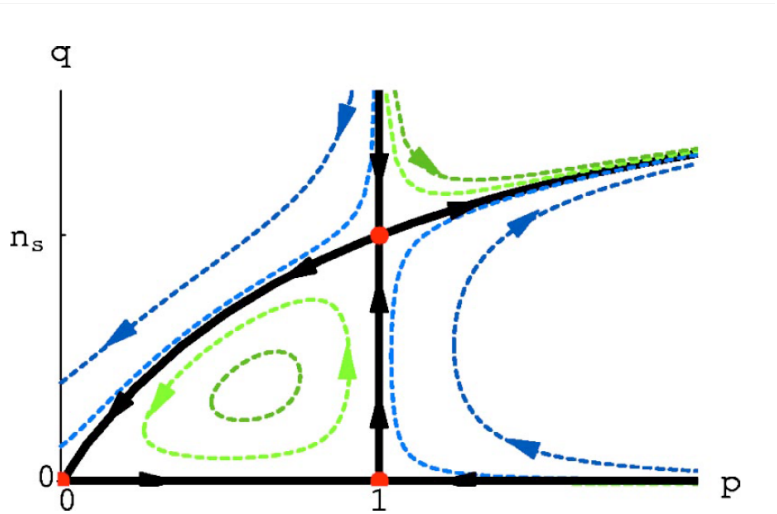
- Red line $n_s = 10$, green $n_s = 50$, blue $n_s = 100$, black $n_S = 1000$

[2] V. Elgart & A. Kamenev. Phys. Rev. E. **70**, 041106 (2004)

Example: Branching with binary anihilation[2]

Gillespie SSA sample paths for different values of the carrying capacity $n_s = \sigma/\lambda$





[2] V. Elgart & A. Kamenev. Phys. Rev. E. **70**, 041106 (2004)

Outline

Numerical methods I: Gillespie stochastic simulation algorithm

Numerical Methods II: The τ-leap method

Motivation[3]

- The SSA has a serious drawback: Poor computational performance, in particular when slow and fast processes are considered

[3] D. Gillespie. J. Chem. Phys. **115**, 1716 (2001)

Motivation[3]

- The SSA has a serious drawback: Poor computational performance, in particular when slow and fast processes are considered
  1. The fast processes dominate the behaviour of the waiting time distribution (i.e. the waiting times generated are very small), and therefore very large number of iterations are necessary to cover simulation times relevant to the dynamics of the slower processes

[3] D. Gillespie. J. Chem. Phys. **115**, 1716 (2001)

Motivation[3]

- The SSA has a serious drawback: Poor computational performance, in particular when slow and fast processes are considered
  1. The fast processes dominate the behaviour of the waiting time distribution (i.e. the waiting times generated are very small), and therefore very large number of iterations are necessary to cover simulation times relevant to the dynamics of the slower processes
- To overcome this situation Gillespie proposed a new numerical scheme to speed up performance with respect to his original method: The τ-leap method, where accuracy is traded off in the benefit of performance

[3] D. Gillespie. J. Chem. Phys. **115**, 1716 (2001)

Motivation[3]

- The SSA has a serious drawback: Poor computational performance, in particular when slow and fast processes are considered
  1. The fast processes dominate the behaviour of the waiting time distribution (i.e. the waiting times generated are very small), and therefore very large number of iterations are necessary to cover simulation times relevant to the dynamics of the slower processes
- To overcome this situation Gillespie proposed a new numerical scheme to speed up performance with respect to his original method: The τ-leap method, where accuracy is traded off in the benefit of performance
- The τ-leap method differs from the SSA in one significant aspect: Whereas the latter involves generating individual events, the former is based on, upon prescription of a time step τ, estimating the number of occurrences of each elementary event durin the time interval $(t, t + \tau)$.

[3] D. Gillespie. J. Chem. Phys. **115**, 1716 (2001)

Derivation of the method I

- We can represent the process $X(t)$ in the following way:

$$X(t) = X_0 + \sum_{i=1}^{R} r_i Z_i(t),$$

where X_0 is the initial condition and $Z_i(t)$ is the advancement coordinate for process i, i.e. the number of times process i has occurred in the interval $(0, t)$

Derivation of the method I

- We can represent the process $X(t)$ in the following way:

$$X(t) = X_0 + \sum_{i=1}^{R} r_i Z_i(t),$$

where X_0 is the initial condition and $Z_i(t)$ is the advancement coordinate for process i, i.e. the number of times process i has occurred in the interval $(0, t)$

- Similarly,

$$X(t+\tau) = X(t) + \sum_{i=1}^{R} r_i Z_i(\tau),$$

where $Z_i(\tau)$ is the number of times process i has occurred in the interval $(t, t+\tau)$

Derivation of the method I

- We can represent the process $X(t)$ in the following way:

$$X(t) = X_0 + \sum_{i=1}^{R} r_i Z_i(t),$$

  where X_0 is the initial condition and $Z_i(t)$ is the advancement coordinate for process i, i.e. the number of times process i has occurred in the interval $(0, t)$

- Similarly,

$$X(t+\tau) = X(t) + \sum_{i=1}^{R} r_i Z_i(\tau),$$

  where $Z_i(\tau)$ is the number of times process i has occurred in the interval $(t, t+\tau)$

- $Z_i(\tau)$ is distributed according to a Poisson distribution with parameter $\lambda_i(t) = \int_0^t W_i(X(s))ds$, i.e.

$$P(Z_i(t) = z) = \frac{(\lambda_i(t))^z}{z!} e^{-\lambda_i(t)}$$

Derivation of the method I

- We can represent the process $X(t)$ in the following way:

$$X(t) = X_0 + \sum_{i=1}^{R} r_i Z_i(t),$$

  where X_0 is the initial condition and $Z_i(t)$ is the advancement coordinate for process i, i.e. the number of times process i has occurred in the interval $(0, t)$

- Similarly,

$$X(t+\tau) = X(t) + \sum_{i=1}^{R} r_i Z_i(\tau),$$

  where $Z_i(\tau)$ is the number of times process i has occurred in the interval $(t, t+\tau)$

- $Z_i(\tau)$ is distributed according to a Poisson distribution with parameter $\lambda_i(t) = \int_0^t W_i(X(s))ds$, i.e.

$$P(Z_i(t) = z) = \frac{(\lambda_i(t))^z}{z!} e^{-\lambda_i(t)}$$

- So, $Z_i(\tau) = \text{Poisson}(\lambda_i(\tau)) \equiv Y_i(\lambda_i(\tau))$

Derivation of the method II

- Hence,

$$X(t+\tau) = X(t) + \sum_{i=1}^{R} r_i Y_i(\lambda_i(\tau)),$$

Derivation of the method II

- Hence,

$$X(t+\tau) = X(t) + \sum_{i=1}^{R} r_i Y_i(\lambda_i(\tau)),$$

- Now, if τ is small, $\lambda_i(\tau)$ can be approximated by:

$$\lambda_i(\tau) = \int_t^{t+\tau} W_i(X(s))ds \simeq W_i(X(t))\tau$$

Derivation of the method II

- Hence,

$$X(t+\tau) = X(t) + \sum_{i=1}^{R} r_i Y_i(\lambda_i(\tau)),$$

- Now, if τ is small, $\lambda_i(\tau)$ can be approximated by:

$$\lambda_i(\tau) = \int_t^{t+\tau} W_i(X(s))ds \simeq W_i(X(t))\tau$$

- From the two equations above, we obtain the τ-leap formula:

$$X(t+\tau) \simeq X(t) + \sum_{i=1}^{R} r_i Y_i(W_i(X(t))\tau)$$

Caveats

Remark 1

τ-leap method$\rightarrow$ SSA when $\tau \rightarrow 0$

Caveats

Remark 1

τ-leap method$\rightarrow$ SSA when $\tau \rightarrow 0$

Remark 2

For the τ-leap formula to be a good approximation, τ and $W_i(X(t))$ must be such that the propensity functions will not suffer and *appreciable* change when $X(t) \rightarrow X(t+\tau) = X(t) + \Delta X_\tau$. This statement, which will be made more precise in the next slide, is the so-called leap condition

Caveats

Remark 1

τ-leap method$\rightarrow$ SSA when $\tau \rightarrow 0$

Remark 2

For the τ-leap formula to be a good approximation, τ and $W_i(X(t))$ must be such that the propensity functions will not suffer and *appreciable* change when $X(t) \rightarrow X(t+\tau) = X(t) + \Delta X_\tau$. This statement, which will be made more precise in the next slide, is the so-called leap condition

Remark 3

If τ is nor chosen properly (i.e. too big), the τ-leap formula can yield negative (unphysical) values of $X(t)$

Leap condition I

- Remarks 2 and 3 imply that chosing τ in a proper manner is critical for the method to produce accurate results

[4] D. Gillespie. J. Chem. Phys. **115**, 1716 (2001)
[5] Y. Cao, D. Gillespie, L.R. Petzold. J Chem. Phys. **124**, 044109 (2006)

Leap condition I

- Remarks 2 and 3 imply that chosing τ in a proper manner is critical for the method to produce accurate results
- There are several methods in the literature. All of their derivations are heuristic and no proof of optimality has been given for any them

[4] D. Gillespie. J. Chem. Phys. **115**, 1716 (2001)
[5] Y. Cao, D. Gillespie, L.R. Petzold. J Chem. Phys. **124**, 044109 (2006)

Leap condition I

- Remarks 2 and 3 imply that chosing τ in a proper manner is critical for the method to produce accurate results
- There are several methods in the literature. All of their derivations are heuristic and no proof of optimality has been given for any them
- We thus focus here on the simplest one, due to Gillespie[4], which is a straightforward application of the leap condition. An improved result has been derived by Cao et al.[5]

[4] D. Gillespie. J. Chem. Phys. **115**, 1716 (2001)
[5] Y. Cao, D. Gillespie, L.R. Petzold. J Chem. Phys. **124**, 044109 (2006)

Leap condition II

- Recall $Z_i(\tau) \simeq Y_i(W_i(X(t))\tau)$

[6] D. Gillespie. J. Chem. Phys. **115**, 1716 (2001)

Leap condition II

- Recall $Z_i(\tau) \simeq Y_i(W_i(X(t))\tau)$
- The expected net change of the state of the system between $(t, t+\tau)$, $\langle \Delta X_\tau \rangle$, is therefore given by:

$$\langle \Delta X_\tau \rangle = \sum_{i=1}^{R} r_i \langle Y_i(W_i(X(t))\tau) \rangle = \sum_{i=1}^{R} r_i W_i(X(t))\tau \equiv \tau \xi(X(t)),$$

where $\xi(X(t))$ is the average (expected) state change per unit time

[6] D. Gillespie. J. Chem. Phys. **115**, 1716 (2001)

Leap condition II

- Recall $Z_i(\tau) \simeq Y_i(W_i(X(t))\tau)$
- The expected net change of the state of the system between $(t, t+\tau)$, $\langle \Delta X_\tau \rangle$, is therefore given by:

$$\langle \Delta X_\tau \rangle = \sum_{i=1}^{R} r_i \langle Y_i(W_i(X(t))\tau) \rangle = \sum_{i=1}^{R} r_i W_i(X(t))\tau \equiv \tau \xi(X(t)),$$

  where $\xi(X(t))$ is the average (expected) state change per unit time
- Furthermore, the leap condition can be stated as $|W_(X(t+\tau)) - W_i(X(t))| \leq \epsilon_i(X(t))$

[6] D. Gillespie. J. Chem. Phys. **115**, 1716 (2001)

Leap condition II

- Recall $Z_i(\tau) \simeq Y_i(W_i(X(t))\tau)$
- The expected net change of the state of the system between $(t, t+\tau)$, $\langle \Delta X_\tau \rangle$, is therefore given by:

$$\langle \Delta X_\tau \rangle = \sum_{i=1}^{R} r_i \langle Y_i(W_i(X(t))\tau) \rangle = \sum_{i=1}^{R} r_i W_i(X(t))\tau \equiv \tau \xi(X(t)),$$

  where $\xi(X(t))$ is the average (expected) state change per unit time
- Furthermore, the leap condition can be stated as $|W_(X(t+\tau)) - W_i(X(t))| \leq \epsilon_i(X(t))$
- Gillespie's criterion[6] consists of taking $X(t+\tau) = X(t) + \langle \Delta X_\tau \rangle$ and $\epsilon_i(X(t)) = \epsilon_0 W_0(X(t))$ where $W_0(X(t)) = \sum_i W_i(X(t))$, i.e. that the variation in $X(t)$ is of the order of the average and that the variation in the propensities are all bound by the inverse of the average waiting time $W_0(X(t)) = \sum_i W_i(X(t))$

[6] D. Gillespie. J. Chem. Phys. **115**, 1716 (2001)

Leap condition III

- By assuming $\frac{\langle \Delta X_\tau \rangle}{X(t)} < 1$, we can write:

$$W_i(X(t) + \langle \Delta X_\tau \rangle) - W_i(X(t)) \simeq \langle \Delta X_\tau \rangle \partial_X W_i(X(t)) = \tau \xi(X(t)) \partial_X W_i(X(t))$$

Leap condition III

- By assuming $\frac{\langle \Delta X_\tau \rangle}{X(t)} < 1$, we can write:

$$W_i(X(t) + \langle \Delta X_\tau \rangle) - W_i(X(t)) \simeq \langle \Delta X_\tau \rangle \partial_X W_i(X(t)) = \tau \xi(X(t)) \partial_X W_i(X(t))$$

- Therefore,

$$\tau \left| \xi(X(t)) \partial_X W_i(X(t)) \right| \leq \epsilon_0 W_0(X(t))$$

Leap condition III

- By assuming $\frac{\langle \Delta X_\tau \rangle}{X(t)} < 1$, we can write:

$$W_i(X(t) + \langle \Delta X_\tau \rangle) - W_i(X(t)) \simeq \langle \Delta X_\tau \rangle \partial_X W_i(X(t)) = \tau \xi(X(t)) \partial_X W_i(X(t))$$

- Therefore,

$$\tau \left| \xi(X(t)) \partial_X W_i(X(t)) \right| \leq \epsilon_0 W_0(X(t))$$

- Gillespie's leap condition:

$$\tau = \min_{i \in [1,M]} \frac{\epsilon_0 W_0(X(t))}{\left| \xi(X(t)) \partial_X W_i(X(t)) \right|}$$

τ-leap algorithm

1. Initialisation $X(t=0) = X_0$. Initialise seed of random number generator. Fix value of ϵ_0

τ-leap algorithm

1. Initialisation $X(t=0) = X_0$. Initialise seed of random number generator. Fix value of ϵ_0
2. Calculate $W_i(X(t))$ for $i = 1, \ldots, R$. Calculate $W_0(X(t)) = \sum_{i=1}^{R} W_i(X(t))$

τ-leap algorithm

1. Initialisation $X(t=0) = X_0$. Initialise seed of random number generator. Fix value of ϵ_0
2. Calculate $W_i(X(t))$ for $i = 1, \ldots, R$. Calculate $W_0(X(t)) = \sum_{i=1}^{R} W_i(X(t))$
3. Calculate the value of τ according to the leap condition

τ-leap algorithm

1. Initialisation $X(t=0) = X_0$. Initialise seed of random number generator. Fix value of ϵ_0
2. Calculate $W_i(X(t))$ for $i = 1, \ldots, R$. Calculate $W_0(X(t)) = \sum_{i=1}^{R} W_i(X(t))$
3. Calculate the value of τ according to the leap condition
4. Generate M random Poisson distributed numbers $Y_i(W_i(X(t))\tau)$

τ-leap algorithm

1. Initialisation $X(t=0) = X_0$. Initialise seed of random number generator. Fix value of ϵ_0
2. Calculate $W_i(X(t))$ for $i = 1, \ldots, R$. Calculate $W_0(X(t)) = \sum_{i=1}^{R} W_i(X(t))$
3. Calculate the value of τ according to the leap condition
4. Generate M random Poisson distributed numbers $Y_i(W_i(X(t))\tau)$
5. Update $X(t+\tau)$ according to the τ-leap formula

τ-leap algorithm

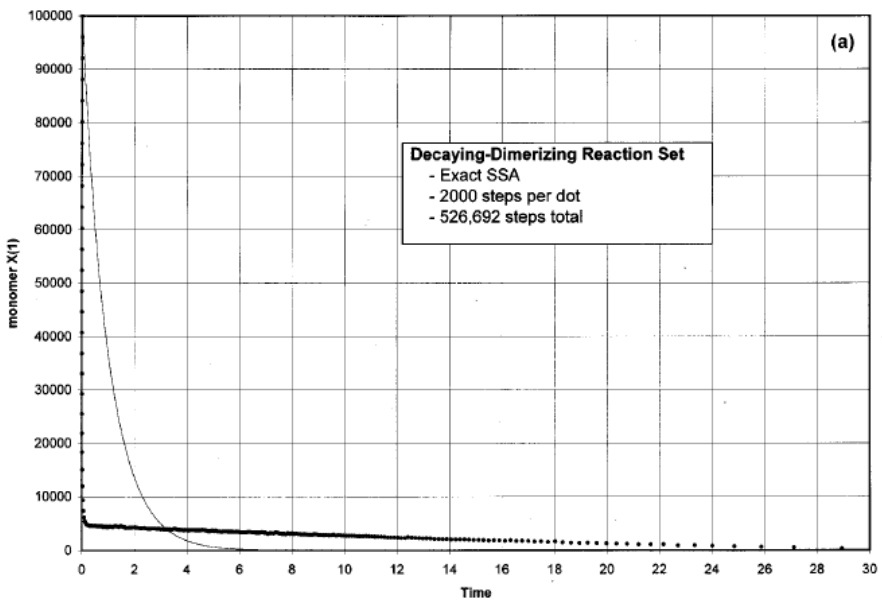
1. Initialisation $X(t=0) = X_0$. Initialise seed of random number generator. Fix value of ϵ_0
2. Calculate $W_i(X(t))$ for $i = 1, \ldots, R$. Calculate $W_0(X(t)) = \sum_{i=1}^{R} W_i(X(t))$
3. Calculate the value of τ according to the leap condition
4. Generate M random Poisson distributed numbers $Y_i(W_i(X(t))\tau)$
5. Update $X(t+\tau)$ according to the τ-leap formula
6. Update $t \leftarrow t + \tau$

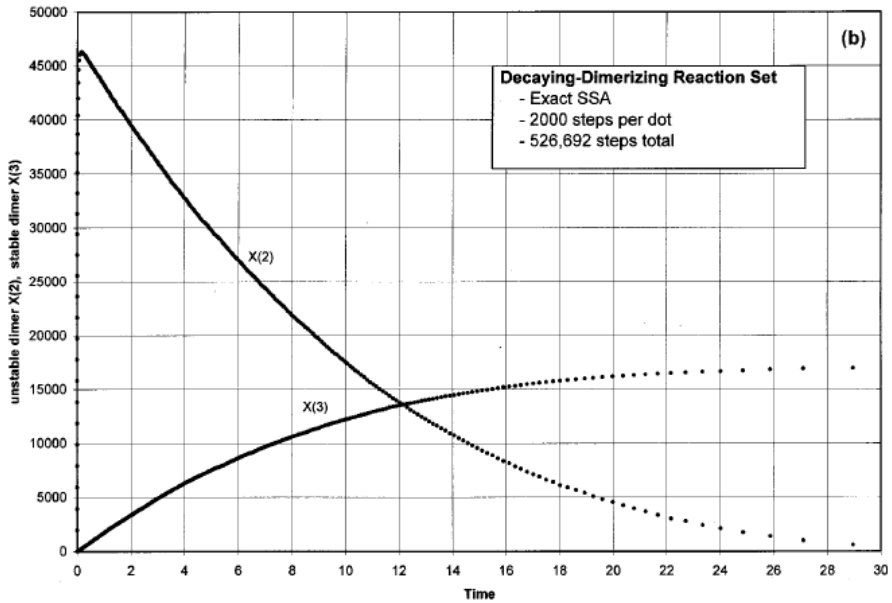
τ-leap algorithm

1. Initialisation $X(t=0) = X_0$. Initialise seed of random number generator. Fix value of ϵ_0
2. Calculate $W_i(X(t))$ for $i = 1, \ldots, R$. Calculate $W_0(X(t)) = \sum_{i=1}^{R} W_i(X(t))$
3. Calculate the value of τ according to the leap condition
4. Generate M random Poisson distributed numbers $Y_i(W_i(X(t))\tau)$
5. Update $X(t+\tau)$ according to the τ-leap formula
6. Update $t \leftarrow t + \tau$
7. Iterate steps 2–6 until some stopping criterion is fulfilled (e.g. $t \geq T$)

Example: Decaying dimerisation[7]

$S_1 \to \varnothing$, $S_1 + S_1 \leftrightarrows S_2$, $S_2 \to S_3$. SSA results

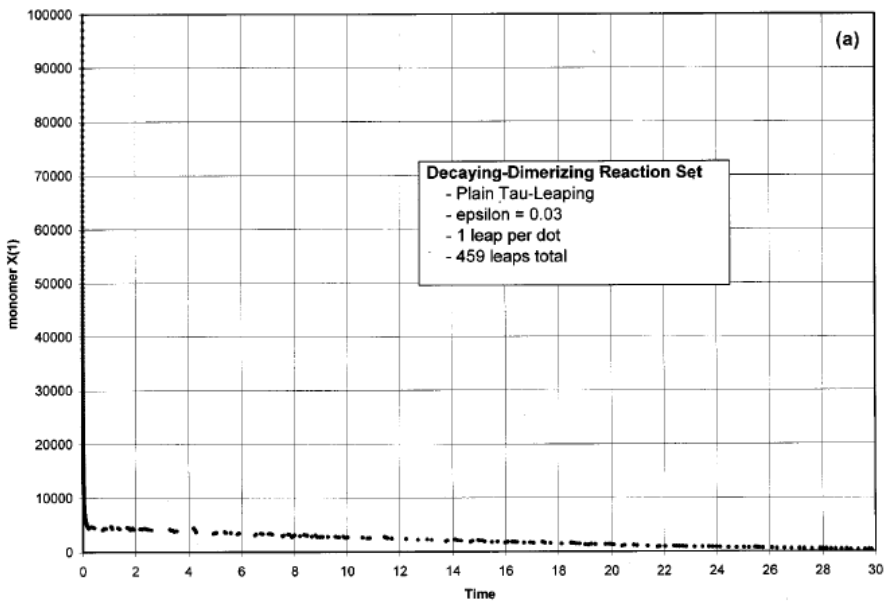


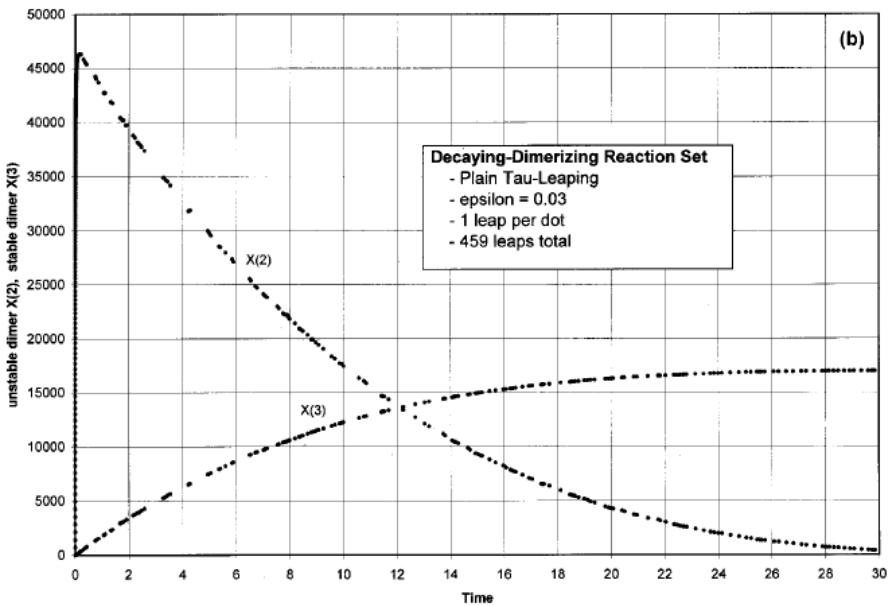


[7] D. Gillespie. J. Chem. Phys. **115**, 1716 (2001)

Example: Decaying dimerisation[8]

$S_1 \to \varnothing$, $S_1 + S_1 \leftrightarrows S_2$, $S_2 \to S_3$. τ-leap algorithm, $\epsilon_0 = 0.03$





[8] D. Gillespie. J. Chem. Phys. **115**, 1716 (2001)

Outline of next lecture

1. Some examples from my own research