WORKSHOP: "STOCHASTIC SIMULATIONS IN BRANCHING PROCESSES"

ERNESTO BERRÍOS-CARO

FÉLIX GEOFFROY

STOCHASTIC EVOLUTIONARY DYNAMICS | MAX PLANCK INSTITUTE FOR EVOLUTIONARY BIOLOGY

τ-LEAPING ALGORITHM

JOURNAL OF CHEMICAL PHYSICS VOLUME 115, NUMBER 4 22 JULY 2001

Approximate accelerated stochastic simulation of chemically reacting systems

Daniel T. Gillespie[a)]
Research Department, Code 4T4100D, Naval Air Warfare Center, China Lake, California 93555

(Received 29 December 2000; accepted 19 April 2001)

The stochastic simulation algorithm (SSA) is an essentially exact procedure for numerically simulating the time evolution of a well-stirred chemically reacting system. Despite recent major improvements in the efficiency of the SSA, its drawback remains the great amount of computer time that is often required to simulate a desired amount of system time. Presented here is the "τ-leap" method, an approximate procedure that in some circumstances can produce significant gains in simulation speed with acceptable losses in accuracy. Some primitive strategies for control parameter selection and error mitigation for the τ-leap method are described, and simulation results for two simple model systems are exhibited. With further refinement, the τ-leap method should provide a viable way of segueing from the exact SSA to the approximate chemical Langevin equation, and thence to the conventional deterministic reaction rate equation, as the system size becomes larger.

RECAP OF NOTATION AND VARIABLES

- For each **reaction** i with, we define their **propensities** as a_i.
  They represent the **probability per unit time** that reaction i will trigger

e.g. $X \xrightarrow{\gamma_i} Y$ $\quad a_i = \gamma_i \cdot n_X$

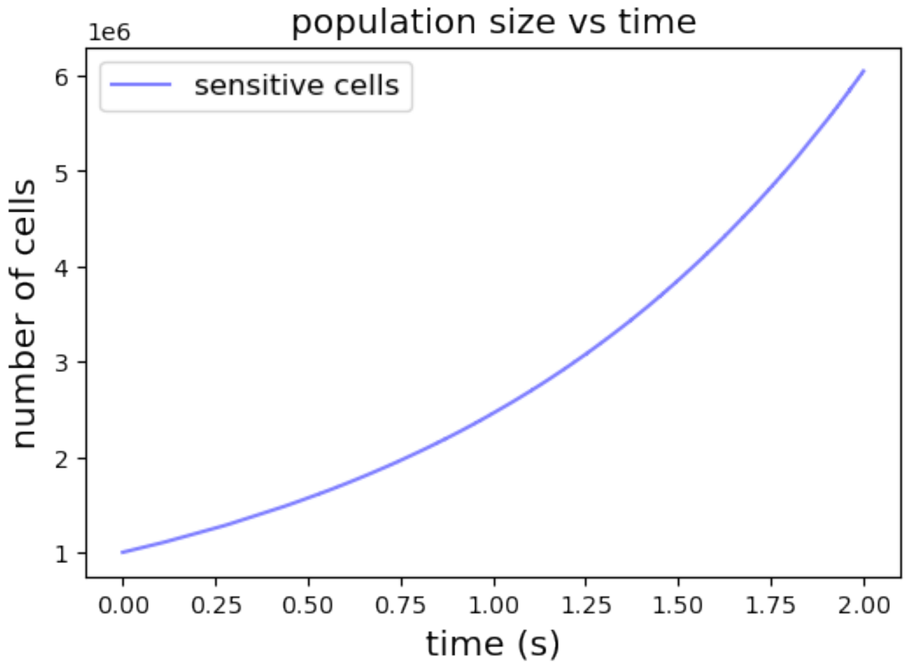
- We define the sum of the propensities of M reactions as $a_0 = \sum_{i=1}^{M} a_i$

- Gillespie algorithm: **time next reaction** $\tau \sim \text{Exp}(a_0)$

  probability that **next event** is reaction i is $\frac{a_i}{a_0}$

LIMITATIONS OF THE GILLESPIE ALGORITHM

- If many reactions trigger per unit time, then the Gillespie algorithm becomes inefficient. This is usually the case for **large population sizes**




```
n =  1000322 t =  0.00037033673594721314
n =  1000323 t =  0.0003708688856994301
n =  1000324 t =  0.000371116019451837
n =  1000323 t =  0.00037218978041237755
n =  1000324 t =  0.00037261119767315856
n =  1000325 t =  0.00037486625817380824
n =  1000326 t =  0.0003768058880812528
n =  1000327 t =  0.00037694998738733044
n =  1000328 t =  0.000377768394432059
n =  1000329 t =  0.00037788535618918584
n =  1000330 t =  0.0003780028262777252
n =  1000331 t =  0.00037888104494180784
n =  1000332 t =  0.0003792932514395348 3
```

τ-LEAP METHOD

- Idea:
  - Choose a **constant time interval** τ.
  - Calculate the **number of times** that each reaction triggers between t and $t + \tau$.
  - Update the system variables **accordingly**.

HOW TO IMPLEMENT THIS IDEA?

- Let us define

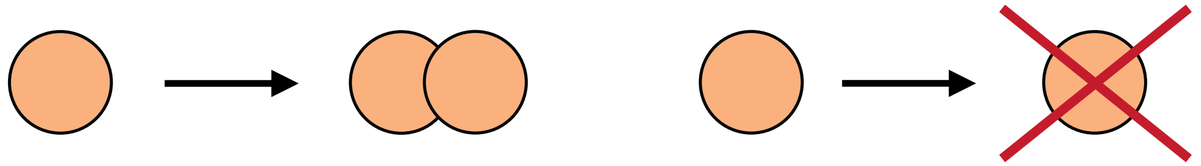
$K_i(\tau)$: **number of times** that event i will fire between t and $t + \tau$

- **Leap condition:** require τ to be small enough so that all the propensities a_i will not suffer a **considerable change** between t and $t + \tau$.

- If the leap condition is fulfilled, then it holds that

$$K_i(\tau) \sim \text{Poisson}(a_i \tau)$$

- **Example**: birth-death process



reaction 1

$$X \xrightarrow{b} X + X$$

$$a_1 = b \cdot n_X$$

reaction 2

$$X \xrightarrow{d} \emptyset$$

$$a_2 = d \cdot n_X$$

- Let $n_X(t)$ be the **number of cells** at time t
- Let's choose a time interval τ that fulfils the **leap-condition**
- **Number of birth events** between $t + \tau$: $K_1(\tau) \sim \text{Poisson}(a_1\tau)$
- **Number of death events** between $t + \tau$: $K_2(\tau) \sim \text{Poisson}(a_2\tau)$
- **Number of cells** at $t + \tau$: $n_X(t + \tau) = n_X(t) + K_1(\tau) - K_2(\tau)$

τ-LEAPING ALGORITHM

1. Initialise the system at $t = 0$. Choose a value for τ that fulfils the **leap condition**.

2. Compute the **propensities** of each reaction a_i at time t. Calculate the **number of times that each reaction i triggers** between t and $t + \tau$ as:

$$K_i(\tau) \sim \text{Poisson}(a_i \tau)$$

3. Denoting by ν_{ij} the **change in n_j caused by reaction i**, update the system as:

$$n_j(t + \tau) = n_j(t) + \sum_i K_i(\tau) \cdot \nu_{ij}$$

4. Update the time at $t + \tau$. Go to 2.

SELECTING τ

- The τ-leaping algorithm requires to choose an **appropriate value for** τ.
- Intuition for choosing τ: the propensities do not change abruptly between t and $t + \tau$
- If τ is too large -> simulations lose precision
- If τ is too small -> simulations become costly and inefficient

- Formal procedure for selecting τ:

- Let us define:

ν_{ij}: the **change in n_j caused by reaction i**

$$\xi_j = \sum_{i=1}^{M} a_i \cdot \nu_{ij}$$

$$b_{ij} = \frac{\partial a_i}{\partial n_j}$$

- Choose τ as:

$$\tau = \min_{i \in [1,M]} \frac{\epsilon a_0}{\sum_{j=1}^{M} \xi_j \cdot b_{ij}}$$

for some $0 < \epsilon < 1$

- For the Gillespie algorithm, the expected time for the next reaction is

$$\tau = \frac{1}{a_0} \qquad \text{with} \qquad a_0 = \sum_{i=1}^{M} a_i$$

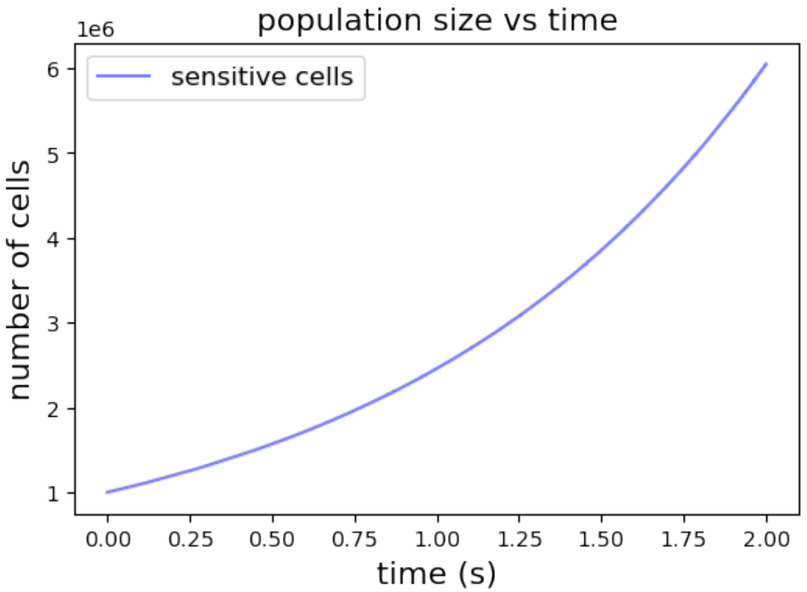
- If the τ chosen for the τ-leaping algorithm is lower than this, it is better to use the Gillespie algorithm

EXAMPLES

$$b_0 = 1.0$$
$$d_0 = 0.1$$
$$n_0 = 10^6$$

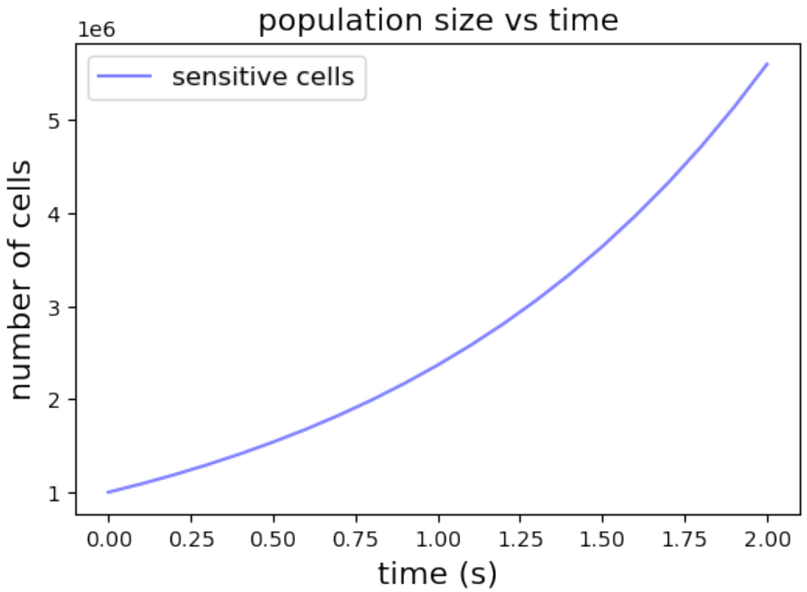
- Birth-death process

Gillespie algorithm



~29.51 seconds

τ-leaping algorithm



~0.11 seconds

EXTENSIONS OF THE τ-LEAPING ALGORITHM

PHYSICAL REVIEW E **104**, 014122 (2021)

Beyond the adiabatic limit in systems with fast environments: A τ-leaping algorithm

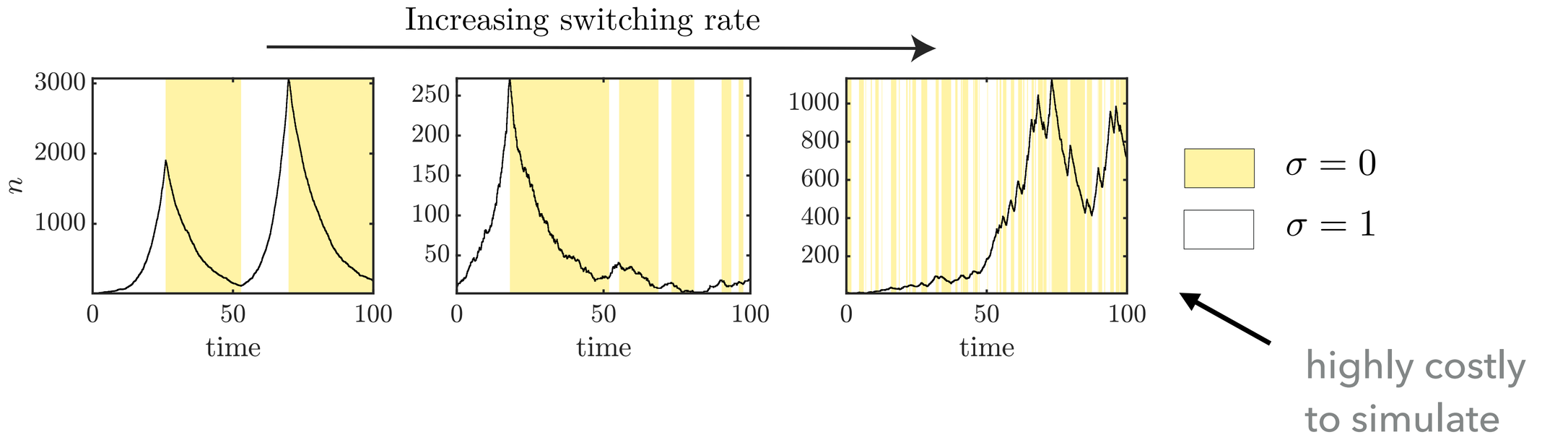
Ernesto Berríos-Caro[1,*] and Tobias Galla[1,2,†]

[1]*Theoretical Physics, Department of Physics and Astronomy, School of Natural Sciences, Faculty of Science and Engineering, The University of Manchester, Manchester M13 9PL, United Kingdom*

[2]*Instituto de Física Interdisciplinar y Sistemas Complejos, IFISC (CSIC-UIB), Campus Universitat Illes Balears, E-07122 Palma de Mallorca, Spain*

(Received 21 November 2020; revised 17 May 2021; accepted 21 June 2021; published 19 July 2021)

MOTIVATION



- In the **infinitely fast switching limit** (adiabatic limit), one can find analytical solutions for relevant stochastic quantities (e.g., probability distributions).
- But what about the "**not too fast switching limit**"?

IDEA OF THE τFE ALGORITHM

- Calculate effective rates that incorporate the effect of the fast switching environment, **without having to simulate the environment**.

birth-death process, 2 species, 3 environments

$$\emptyset \xrightarrow{\Omega\alpha_\sigma} A, \quad \emptyset \xrightarrow{\Omega\beta_\sigma} B,$$
$$A \xrightarrow{d_A} \emptyset, \quad B \xrightarrow{d_B} \emptyset,$$
$$E_0 \xrightarrow{\lambda k_1} E_1, \quad E_1 \xrightarrow{\lambda k_2} E_2, \quad E_2 \xrightarrow{\lambda k_0} E_0$$

effective system

$$\emptyset \xrightarrow{\Omega\bar{\alpha}} A, \quad \emptyset \xrightarrow{\Omega\bar{\beta}} B,$$
$$A \xrightarrow{d_A} \emptyset, \quad B \xrightarrow{d_B} \emptyset$$

the rates $\bar{\alpha}$ and $\bar{\beta}$ incorporate the effect of the fast switching environment

- Then, we simulate the effective system using the τ-leaping

$$\emptyset \xrightarrow{\Omega\bar{\alpha}} A, \quad \emptyset \xrightarrow{\Omega\bar{\beta}} B,$$
$$A \xrightarrow{d_A} \emptyset, \quad B \xrightarrow{d_B} \emptyset$$

- The rates $\bar{\alpha}$ and $\bar{\beta}$ are correlated Gaussian random numbers with mean:

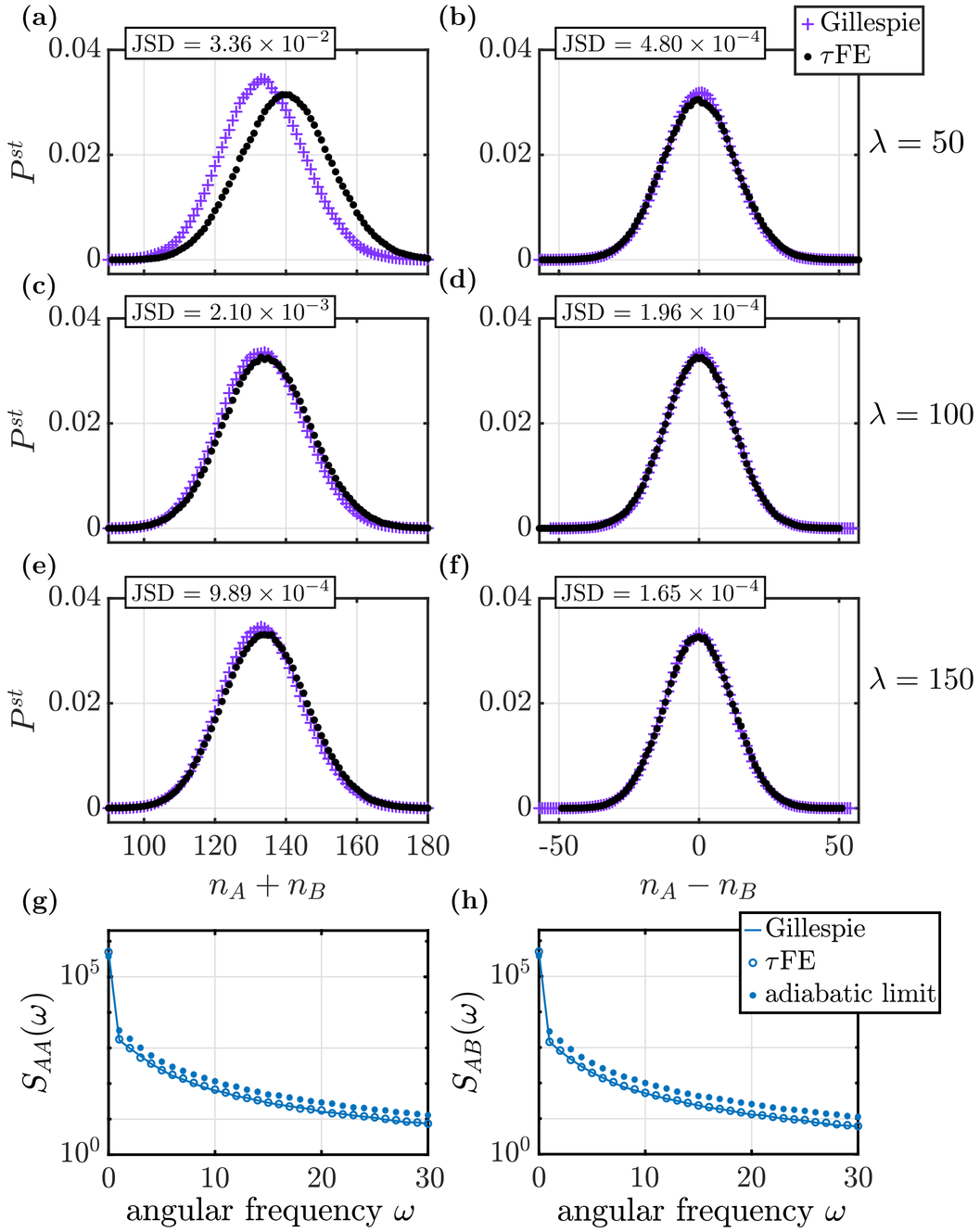
$$\alpha^* = \frac{k_0\alpha_0 + k_1\alpha_1 + k_2\alpha_2}{k_0 + k_1 + k_2}$$
$$\beta^* = \frac{k_0\beta_0 + k_1\beta_1 + k_2\beta_2}{k_0 + k_1 + k_2}$$

and covariance:

$$\sigma_{\alpha\beta} = \frac{\theta^2}{\lambda\Delta t}\Big\{(\alpha_0 - \alpha_1)(\beta_0 - \beta_1)\left(3k_0^2 - k_{0,1}k_{0,2}\right) + (\alpha_1 - \alpha_2)(\beta_1 - \beta_2)\left(3k_1^2 - k_{1,0}k_{1,2}\right) + (\alpha_0 - \alpha_2)(\beta_0 - \beta_2)\left(3k_2^2 - k_{2,0}k_{2,1}\right)\Big\}$$

in each iteration we draw $\bar{\alpha}$ and $\bar{\beta}$

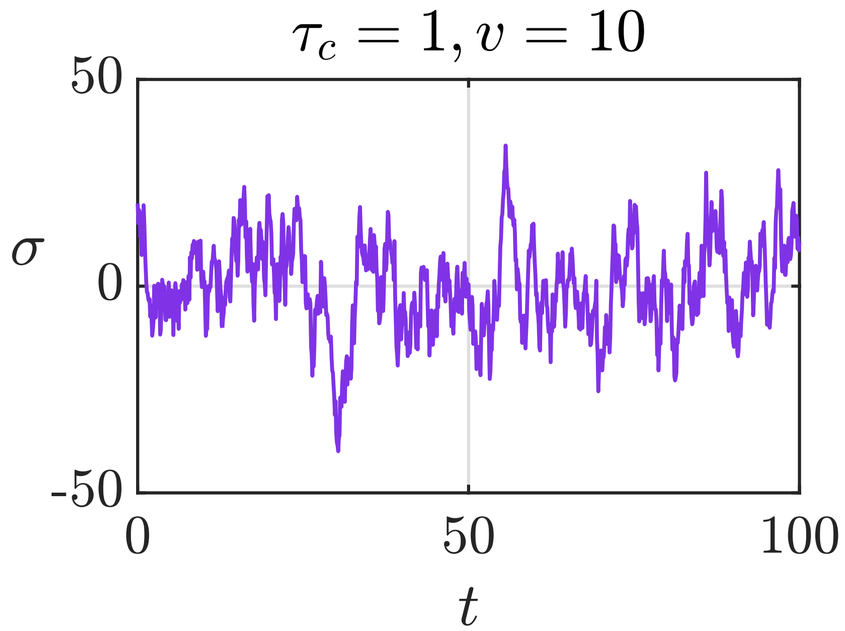
environmental
switching rate



the τFE algorithm can be ~100 times faster than the Gillespie algorithm

CONTINUOUS ENVIRONMENTS

- Analogous idea but with **continuous environmental state** σ

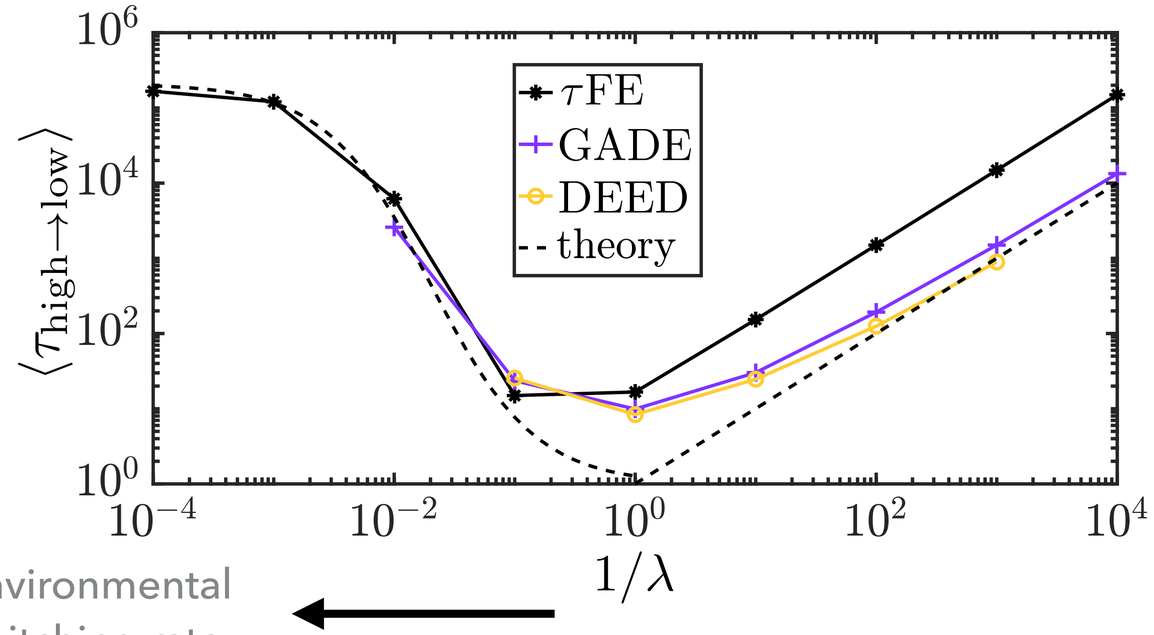


we assumed σ followed an **Ornstein-Uhlenbeck process**

- Example: genetic switch with Hill-like regulatory function

$$f(x,\sigma) = \alpha_0 + (1 - \alpha_0 + \sigma)\Theta(x - x_0)$$

production rate of proteins



environmental switching rate

λ^{-1}	GADE	DEED	τFE
1×10^{-2}	28.47	3.84	0.79×10^{-2}
5×10^{-3}	53.20	7.88	0.16×10^{-1}
1×10^{-3}	288.30	40.91	0.08
5×10^{-4}	576.69	82.78	0.15
1×10^{-4}	3022.47	397.63	0.79

GADE: Gillespie algorithm with discretised environmental states

DEED: Discrete-time simulation with explicit environmental dynamics