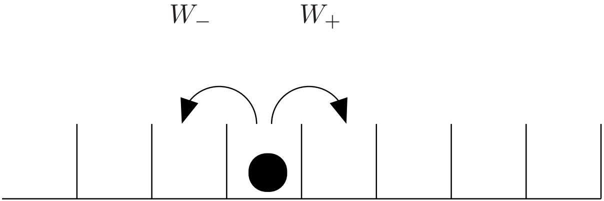
Numerical algorithms for the evaluation of large-deviation functions

AI

Motivations

- Simulation of systems with large number of microscopic states
- Simulation of systems with complex (stochastic) dynamical equation
- Evaluation of the probability of rare events through the cumulant generating function

Random walk on 1d-lattice



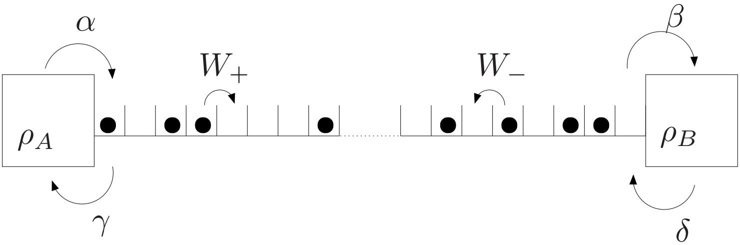
$P_l(t)$ probability that the particle is at site l at time t

$$\frac{\mathrm{d}P_l(t)}{\mathrm{d}t} = W_+P_{l-1} + W_-P_{l+1} - (W_+ + W_-)P_l = J_{l-1/2} - J_{l+1/2}$$

$$J_{l+1/2} = W_+P_l - W_-P_{l+1}; \qquad J_{l-1/2} = W_+P_{l-1} - W_-P_l$$

ASEP

the asymmetric simple exclusion process (ASEP)



$n_l = 0, 1$ occupation number, $\rho_l(t) \equiv \langle n_l(t) \rangle$

$$\frac{\mathrm{d}\rho_l(t)}{\mathrm{d}t} = \left\langle J_{l-1/2} - J_{l+1/2} \right\rangle$$

$$J_{l+1/2} = W_+ n_l (1 - n_{l+1}) - W_- n_{l+1} (1 - n_l)$$
$$J_{l-1/2} = W_+ n_{l-1} (1 - n_l) - W_- n_l (1 - n_{l-1})$$

Stochastic matrix and biased matrix

- One can also use the master equation
  set $x = (n_1, n_2, \ldots, n_N)$ and assume, for example, that the allowed transitions $x \to x'$ are those between states that differ by only one particle jump
  the off-diagonal elements of the stochastic matrix $\boldsymbol{K}$ will be $W_\pm$
- consider the current $J[\boldsymbol{x}]$ with elementary current elements $J_{x'x}$
  Generating function of the moments of J evolves according to

$$\partial_t \Psi(x,t) = \sum_{x'(\neq x)} k_{x,x'} \mathrm{e}^{sJ_{xx'}} \Psi(x',t) - k_{x',x} \Psi(x,t)$$

Tilted matrix $\boldsymbol{L}(s)$ with elements

$$L_{x,x'}(s) = \begin{cases} \mathrm{e}^{sJ_{xx'}} k_{xx'}, \text{ if } x \neq x' \\ -k_x^{\mathrm{out}}, \quad \text{ if } x = x' \end{cases}$$

$k_x^{\mathrm{out}} = \sum_{x'(\neq x)} k_{x'x}$

For future use

- Let us introduce, for future use, the stochastic matrix $\tilde{\boldsymbol{L}}(s)$

$$\tilde{L}_{x,x'}(s) = \begin{cases} \mathrm{e}^{sJ_{xx'}} k_{xx'}, \text{ if } x \neq x' \\ -\tilde{L}_x^{\text{out}}(s), \text{ if } x = x' \end{cases}$$

with

$$\tilde{L}_x^{\text{out}}(s) = \sum_{x'(\neq x)} k_{x'x} \mathrm{e}^{sJ_{x'x}} = \sum_{x'(\neq x)} \tilde{L}_{x',x}(s)$$

keep in mind that $L_{x'x} = \tilde{L}_{x'x}$ for $x' \neq x$

Numerical simulations, discrete phase space, no bias

- Initialize the system in a reasonable and easy–to prepare initial state
- Discretize the time with n time step τ sufficiently small, such that $\tau k_{x'x} < 1 \ \forall x', x$
- At each time step t_i, given that the system is in the microscopic state $x = (n_1, n_2, \ldots, n_N)$, calculate the number $n^{\text{out}}(x)$ of possible states x' accessible from x by, e.g., a single particle jump.
- choose one of the possible $n^{\text{out}}(x)$ accessible states with uniform probability, and perform the transition $x \to x'$ with rate $k_{x'x}$.
- All in all a move is made with probability $\tau k_{x'x}$, and no move is made with probability $1 - \tau k_x^{\text{out}}$ (some care is needed to avoid spurious contributions to the probability $\tau k_{x'x}/n^{\text{out}}(x)$)
- This problem does not arise in systems without kinetic constraints, e.g., the Ising model
- Iterate

Quantities of interest

In the following we will assume time independent rates $k_{x'x}$

Sample the observable(s) of interest $A(x)$ (or $A(x,t)$) along the trajectories

Suppose that you have waited long enough for the system to reach its steady state at $t \gg 0$. Now reset the timer at t_0 and start to sample the observables along the trajectories $\boldsymbol{x} = (x_0, x_1, \dots x_{N_t})$

$$\begin{aligned} \langle A \rangle &= \sum_{\boldsymbol{x}} A[\boldsymbol{x}] P(x_n, t_n | x_{n-1}, t_{n-1}) P(x_{n-1}, t_{n-1} | x_{n-2}, t_{n-2}) \\ &\quad \dots P(x_1, t_1 | x_0, t_0) p(x_0, t_0) \end{aligned}$$

We shall be interested in

$$\begin{aligned} \left\langle \mathrm{e}^{sJ} \right\rangle = \sum_{\boldsymbol{x}} \quad & \mathrm{e}^{sJ_{x_n x_{n-1}}} P(x_n, t_n | x_{n-1}, t_{n-1}) \\ & \mathrm{e}^{sJ_{x_{n-1} x_{n-2}}} P(x_{n-1}, t_{n-1} | x_{n-2}, t_{n-2}) \\ & \dots \mathrm{e}^{sJ_{x_1 x_0}} P(x_1, t_1 | x_0, t_0) p(x_0, t_0) \\ & \approx \boxed{\mathrm{e}^{t\psi(s)}} \end{aligned}$$

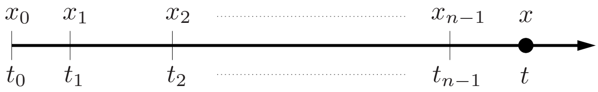
Continuous time framework, no bias

In the continuous time formalism the (non normalized) propagator reads

$$P(x_k, t_k | x_{k-1}, t_{k-1}) = k_{x_k, x_{k-1}} \mathrm{e}^{-k^{\mathrm{out}}_{x_{k-1}}(t_k - t_{k-1})}$$

$$\begin{aligned} \langle A \rangle &= \int \mathcal{D}\boldsymbol{x}\, A[\boldsymbol{x}]\, \mathrm{e}^{-k^{\mathrm{out}}_{x}(t-t_{n-1})} k_{x,x_{n-1}} \mathrm{e}^{-k^{\mathrm{out}}_{x_{n-1}}(t_{n-1}-t_{n-2})} \dots \\ & \quad \dots k_{x_k,x_{k-1}} \mathrm{e}^{-k^{\mathrm{out}}_{x_k}(t_k - t_{k-1})} \dots k_{x_1,x_0} \mathrm{e}^{-k^{\mathrm{out}}_{x_0}(t_1-t_0)} p(x_0, t_0) \end{aligned}$$

Continuous time, no bias : simulations



- Initialize the system in a reasonable and easy–to prepare initial state
- At each time step t_k, given that the system is in the microscopic state x_k draw the waiting time $\tau = t_{k+1} - t_k$ to the next jump from the distribution

$$\rho(x_k, \tau) = k^{\mathrm{out}}_{x_k} \mathrm{e}^{-k^{\mathrm{out}}_{x_k} \tau}$$

- set the timer to $t_{k+1} = \tau + t_k$ and draw the next state x_{k+1} from the probability distribution

$$p(x_k \to x') = k_{x' x_k} / k^{\mathrm{out}}_{x_k}$$

Numerical simulations with bias: cloning algorithm

- we are interested in

$$\langle \mathrm{e}^{sJ} \rangle = \int \mathcal{D}\boldsymbol{x}\, \mathrm{e}^{-k_x^{\text{out}}(t-t_{n-1})} \overbrace{\left(k_{x,x_{n-1}} \mathrm{e}^{sJ_{xx_{n-1}}}\right)}^{L_{xx_{n-1}}} \mathrm{e}^{-k_{x_{n-1}}^{\text{out}}(t_{n-1}-t_{n-2})} \ldots$$

$$\ldots \overbrace{\left(k_{x_k,x_{k-1}} \mathrm{e}^{sJ_{x_k x_{k-1}}}\right)}^{L_{x_k x_{k-1}}} \mathrm{e}^{-k_{x_k}^{\text{out}}(t_k-t_{k-1})}$$

$$\ldots \overbrace{\left(k_{x_1,x_0} \mathrm{e}^{sJ_{x_1 x_0}}\right)}^{L_{x_1 x_0}} \mathrm{e}^{-k_{x_0}^{\text{out}}(t_1-t_0)} p(x_0,t_0) \approx \boxed{\mathrm{e}^{t\psi(s)}}$$

- the stochastic equation for $\Psi(x,\lambda,t)$ depends on the tilted matrix (operator) $\boldsymbol{L}(s)$ and can be solved for a relatively small number of states. Example: for the ASEP with 100 sites, 2^{100} coupled equations.
- Introduce

$$Y_x(\tau,s) = \mathrm{e}^{(\tilde{L}_x^{\text{out}}(s) - k_x^{\text{out}})t}$$

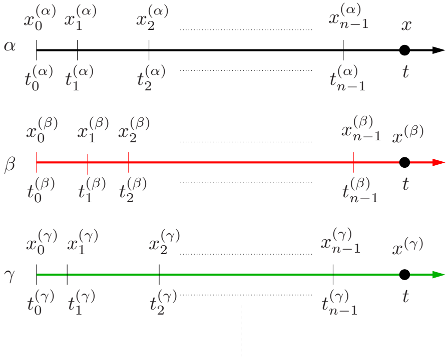
Numerical simulations with bias: cloning algorithm II

- we are interested in

$$
\begin{aligned}
\left\langle \mathrm{e}^{sJ} \right\rangle &= \int \mathcal{D}\boldsymbol{x}\, \mathrm{e}^{-\tilde{L}_x^{\mathrm{out}}(t-t_{n-1})} Y_x(t-t_{n-1}) L_{x x_{n-1}} \mathrm{e}^{-\tilde{L}_{x_{n-1}}^{\mathrm{out}}(t_{n-1}-t_{n-2})} \\
& Y_{x_{n-1}}(t_{n-1}-t_{n-2}) L_{x_{n-1} x_{n-2}} \mathrm{e}^{-\tilde{L}_{x_{n-2}}^{\mathrm{out}}(t_{n-2}-t_{n-3})} \dots \\
& \dots Y_{x_k}(t_k - t_{k-1}) L_{x_k x_{k-1}} \mathrm{e}^{-\tilde{L}_{x_k}^{\mathrm{out}}(t_{k-1}-t_{k-2})} \\
& \dots L_{x_1 x_0} \mathrm{e}^{-\tilde{L}_{x_0}^{\mathrm{out}}(t_1-t_0)} Y_{x_0}(t_1-t_0) p(x_0,t_0) \approx \boxed{\mathrm{e}^{t\psi(s)}}
\end{aligned}
$$

- Now the dynamics is ruled by the matrix $\tilde{\boldsymbol{L}}(s)$ (keep in mind $L_{x'x} = \tilde{L}_{x'x}$ for $x' \neq x$) but there is the new "current" Y
- Solution: introduce M independent clones of the same stochastic system that do not interact and are each described by a master equation with matrix $\tilde{\boldsymbol{L}}(s)$

Numerical simulations with bias: cloning algorithm III



- Start a trajectory from the steady state as obtained, e.g., from the standard algorithm in the initial slides without bias.
- The state of each clones is updated $x_k \to x_{k+1}$ at random time t_k as described above
- Add a cloning step to sample Y

Numerical simulations with bias: cloning algorithm III

- In practice "resize" the population by an amount Y each time a clone jumps.
- example: clone γ immediately before $t_2^{(\gamma)}$
  Copy the clone into approximately $Y_{x_1^{(\gamma)}}(t_2^{(\gamma)} - t_1^{(\gamma)})$ other clones
  Evaluate $y = \lfloor Y_{x_1^{(\gamma)}}(t_2^{(\gamma)} - t_1^{(\gamma)}) + \epsilon \rfloor$, ϵ is a uniformly distributed random number $0 \leq \epsilon < 1$. One has $\langle y \rangle = \lfloor Y_{x_1^{(\gamma)}}(t_2^{(\gamma)} - t_1^{(\gamma)}) \rfloor$
  if $y = 0$, remove the clone.
  if $y > 1$, add $y - 1$ copies of the clone to the population.
- Resize the population to M.
  If $y = 0$, pick up a random clone (of the remaining $M - 1$) and add a copy of it to the population.
  If $y > 1$, pick up at random M clones among the $M + y - 1$ available
- Store the rescaling factor $X = M/(M + y - 1)$

Numerical simulations with bias: a different approach

- the population would grow (or shrink) by a factor $1/X$ without resizing, this corresponds to the dynamics of $\boldsymbol{L}(s)$ not conserving the normalization.
- $$\Psi(s,t) = \left\langle \mathrm{e}^{sJ} \right\rangle \sim (X_1 X_2 \dots X_m)^{-1}$$

  for large time
- At intermediate time, since the dynamics does not preserve normalization, the distribution of clones cannot be interpreted as a conditioned stationary distribution

Application to lattice models

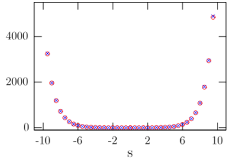


Figure 3. Plot of the large deviation function $(1/L)\psi_Q(s)$ of the asymmetric simple exclusion process, for $L = 400$ sites and $N = 200$ particles. The jump rates are $p = 1.2$ and $q = 0.8$, whence $E \simeq -0.2$. Blue crosses and red circles correspond to direct computation and thermodynamic integration, respectively.

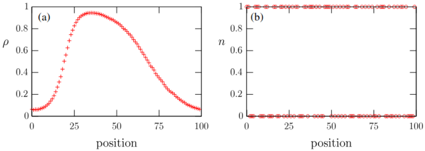


Figure 4. (a) Average profile ρ for $s = 0.3$. To minimize the overall current, the system develops an asymmetric shock, where only the front particles can jump easily. (b) A typical configuration for $s \gg |E|$. The particles are distributed almost uniformly. Note that $|s| \gg |E|$ with $s < 0$ gives a similar result.

From V. Lecomte and J. Tailleur, J. Stat. Mec: Theory and Exp. P03004, 2007.

Numerical simulations with bias: a different approach

- The moment generating function for a given current J is defined as

$$\Psi(s,t) = \sum_x \Psi(x,s,t) = \int \mathcal{D}\boldsymbol{x}\, \mathcal{P}[\boldsymbol{x}]\mathrm{e}^{sJ[\boldsymbol{x}]}$$

- Introduce the new tilted trajectory probability

$$\widehat{\mathcal{P}}[\boldsymbol{x};s] \equiv \mathcal{P}[\boldsymbol{x}]\mathrm{e}^{sJ[\boldsymbol{x}]}/\Psi(s,t)$$

$\Psi(s,t)$ plays the role of a partition function in the ensemble of trajectories with tilted dynamics
Let $\langle A\rangle_s$ be the average of the quantity A over this ensemble of trajectories

$$\begin{aligned}\langle A\rangle_s \quad &= \quad \frac{1}{\Psi(s,t)}\int \mathcal{D}\boldsymbol{x}\mathrm{e}^{-k_x^{\mathrm{out}}(t-t_{n-1})}\left(k_{xx_{n-1}}\mathrm{e}^{sJ_{xx_{n-1}}}\right)\mathrm{e}^{-k_{x_{n-1}}^{\mathrm{out}}(t_{n-1}-t_{n-2})}\cdot\\ &\ldots \quad \left(k_{x_kx_{k-1}}\mathrm{e}^{sJ_{x_kx_{k-1}}}\right)\mathrm{e}^{-k_{x_{k-1}}^{\mathrm{out}}(t_k-t_{k-1})}\\ &\ldots \quad \left(k_{x_1x_0}\mathrm{e}^{sJ_{x_1x_0}}\right)\mathrm{e}^{-k_{x_0}^{\mathrm{out}}(t_1-t_0)}p(x_0,t_0)\cdot A[\boldsymbol{x}]\end{aligned}$$

Numerical simulations with bias: a different approach II

- In particular one has

$$\partial_s \Psi(s,t) = \langle J \rangle_s \Psi(s,t)$$

and thus

$$\Psi(s,t) = \exp\left(\int_0^s ds' \, \langle J \rangle_{s'} \right)$$

- The direct simulation of trajectories in the "s-ensemble" is hindered by the fact that the ensemble partition function, i.e., the function $\Psi(s,t)$ is unknown
In principle can be directly evaluated through simulations for small s to obtain an estimate of $\langle e^{sJ} \rangle$, but unfeasible for large systems and intermediate-to-large $|s|$

Numerical simulations... : a different approach III

- Let us introduce a new functional $\Pi[\boldsymbol{x}]$
  One has

$$\langle J\rangle_s = \frac{\int \mathcal{D}\boldsymbol{x}\, (J[\boldsymbol{x}]\,\Pi[\boldsymbol{x}])\mathcal{P}[\boldsymbol{x}]\Pi^{-1}[\boldsymbol{x}]\mathrm{e}^{sJ[\boldsymbol{x}]}}{\int \mathcal{D}\boldsymbol{x}\,\Pi[\boldsymbol{x}]\mathcal{P}[\boldsymbol{x}]\Pi^{-1}[\boldsymbol{x}]\mathrm{e}^{sJ[\boldsymbol{x}]}} = \frac{\langle J\,\Pi\rangle_{s,\Pi}}{\langle \Pi\rangle_{s,\Pi}}, \qquad \text{(i)}$$

  with

$$\widetilde{\mathcal{P}}[\boldsymbol{x}; s, \Pi] \equiv \frac{\mathcal{P}[\boldsymbol{x}]\Pi^{-1}[\boldsymbol{x}]\mathrm{e}^{sJ[\boldsymbol{x}]}}{\int \mathcal{D}\boldsymbol{x}\,\mathcal{P}[\boldsymbol{x}]\Pi^{-1}[\boldsymbol{x}]\mathrm{e}^{sJ[\boldsymbol{x}]}}, \qquad \text{(ii)}$$

- Now choose

$$\Pi[\boldsymbol{x}] = \prod_{i=0}^{n-1} \mathrm{e}^{(\tilde{L}^{\mathrm{out}}_{x_i}(s) - k^{\mathrm{out}}_{x_i})(t_{i+1}-t_i)}$$

  then, for example, the denominator in (ii) reads

$$\int \mathcal{D}\boldsymbol{x}\mathrm{e}^{-\tilde{L}^{\mathrm{out}}_{x}(t-t_{n-1})} L_{x x_n} \mathrm{e}^{-\tilde{L}^{\mathrm{out}}_{x_{n-1}}(t_{n-1}-t_{n-2})} \dots L_{x_k x_{k-1}} \mathrm{e}^{-\tilde{L}^{\mathrm{out}}_{x_k}(t_k - t_{k-1})}$$

$$\dots L_{x_1 x_0} \mathrm{e}^{-\tilde{L}^{\mathrm{out}}_{x_0}(t_1-t_0)} p(x_0, t_0) = 1$$

Numerical simulations... : a different approach IV

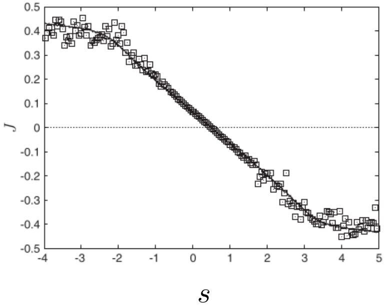
This corresponds to simulate the stochastic dynamics described by the master equation

$$\partial_t \tilde{p}(x,t) = \sum_{x'} \tilde{L}_{x,x'}(s)\tilde{p}(x',t) - \tilde{L}_{x',x}(s)\tilde{p}(x,t)$$

and one can sample the quantities $\langle J\,\Pi \rangle_{s,\Pi}$ and $\langle \Pi \rangle_{s,\Pi}$ in eq. (i) in the previous slide.

Numerical simulations... : a different approach V

ASEP with $L = 100$ sites, $W_+ = 1$, $W_- = 0.75$, $\rho_A = 0.75$, $\rho_B = 0.25$, which correspond to the maximum current phase



average particle current J in the weighted trajectory s-ensemble (full line) and in the (s, Π)-ensemble (squares), as a function of s. The current vanishes at $s = 1/2$, and becomes negative for larger values of s. Total time $t_f = 5$, with the elementary time step $\tau = 0.01$ and 1000 independent trajectories