Ergodicity of Markov processes: theory and computation (4)

Yao Li

Department of Mathematics and Statistics, University of Massachusetts Amherst

September 15, 2021

ICERM, Brown University

Overview

Last three times:

1. Find a small set C
2. Use Lyapunov function to estimate η_C
3. Create an atom α. Estimate η_α
4. Use renewal theory to estimate the coupling time τ_C

Today: Data-driven computing for ergodicity and invariant probability measures.

Why data-driven? Traditional methods do not work in high dimension!

Outline: Data-driven computation

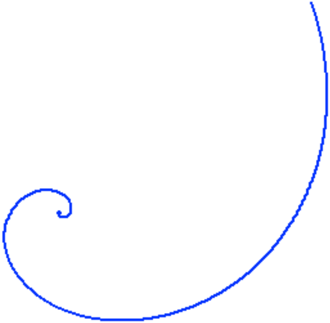
I. Invariant probability measure

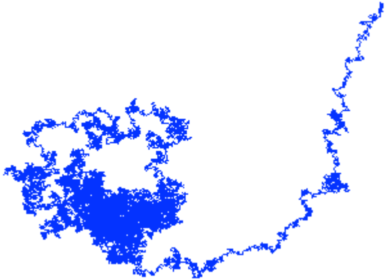
- Combine traditional PDE solver with simulation data.
- Data-driven solver for invariant probability measure.

II. Geometric/power-law ergodicity

- How fast does SDE converge to π?
- Estimate coupling time from data.

Noise perturbations





ODE system

$$\frac{\mathrm{d}X}{\mathrm{d}t} = f(X)$$

stochastic differential equation (SDE)

$$\mathrm{d}X_t = f(X_t)\mathrm{d}t + \epsilon\sigma(x)dW_t\,.$$

Noise perturbations

1. Theory for discrete-time Markov process still works
2. Transition kernel becomes time-dependent

$$P^t(x, A) = \mathbb{P}[\Phi_t \in A \mid \Phi_0 = x]$$

3. Infinitesimal generator

$$\mathcal{L}u = \lim_{t \to 0} \frac{P^t u - u}{t} = \sum_{i=1}^{n} f_i u_{x_i} + \frac{1}{2}\epsilon^2 \sum_{i,j=1}^{n} a_{ij} u_{x_i x_j}$$

where $A = \{a_{ij}\}_{i,j=1}^{d} = \sigma\sigma^T$

Fokker-Planck equation and its steady-state

Invariant probability measure

- Let $P^t(x, \cdot)$ be the transition kernel of the SDE
- A probability measure μ is said to be invariant if

$$\mu(A) = \int \mu(\mathrm{d}x) P^t(x, A) \quad \text{for any measurable } A\,.$$

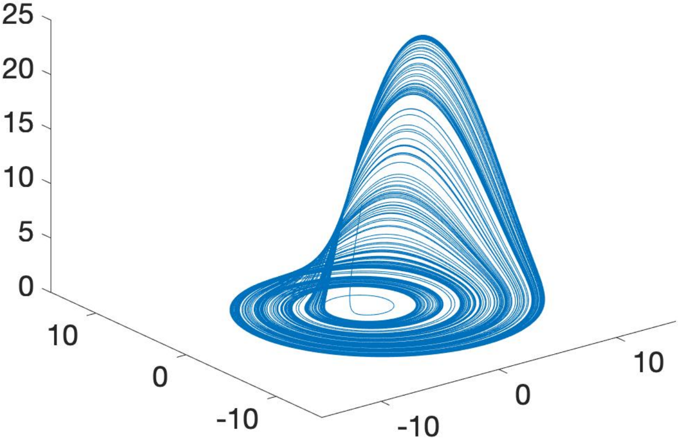
Steady-state Fokker-Planck equation

The density function of μ solves equation

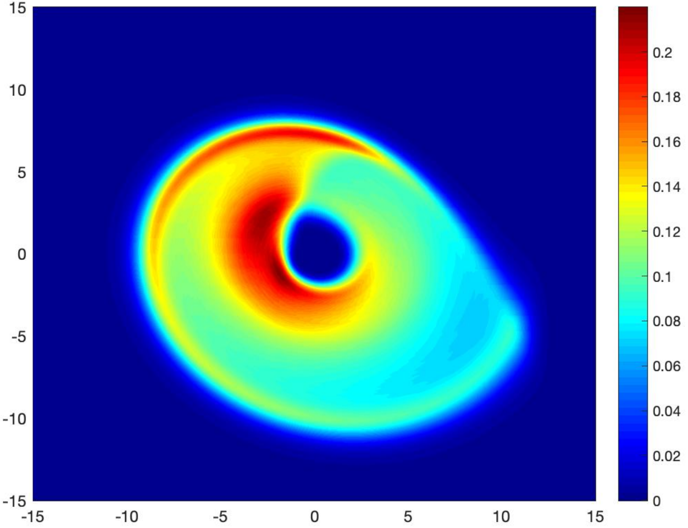
$$\mathcal{L}^* u = \frac{1}{2}\epsilon^2 \sum_{i,j=1}^{N} (a_{ij} u)_{ij} - \nabla \cdot (fu) = 0\,,$$

where $A = \{a_{ij}\}_{i,j=1}^{d} = \sigma\sigma^T$.

Numerical example: Rossler Attractor



Numerical example: Rossler Attractor + noise



How to compute invariant density function?

Numerical PDE approach

Discretize steady-state Fokker-Planck equation

$$\frac{1}{2}\epsilon^2 \sum_{i,j=1}^{N} (a_{ij}u)_{ij} - \nabla \cdot (fu) = 0\,.$$

Problem: What's the boundary condition?

- Sufficiently large numerical domain.
- Use large deviations. Zero boundary.
- Find least square solution.
- High computational cost in general.

How to compute invariant density function?

Monte Carlo method

- Divide the domain into many bins $B_1, \cdots, B_N$.
- Run a long SDE trajectory.
- Count samples in each bin. Estimate density.
- Works for arbitrary numerical domain.

Problem: accuracy

- Not many sample points in each bin. High relative error.
- Solution looks "furry" even with large number of samples.

Numerical example: Gradient flow + rotation

Equation

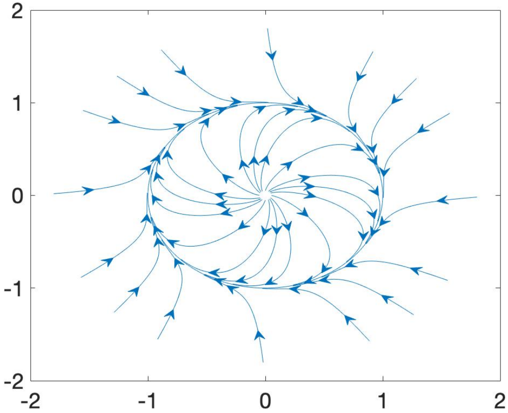
$$\begin{aligned} \mathrm{d}X_t &= (Y_t - 4X_t(X_t^2 + Y_t^2 - 1))\mathrm{d}t + \mathrm{d}W_t \\ \mathrm{d}Y_t &= (-X_t - 4Y_t(X_t^2 + Y_t^2 - 1))\mathrm{d}t + \mathrm{d}W_t \end{aligned}$$

Invariant probability density function

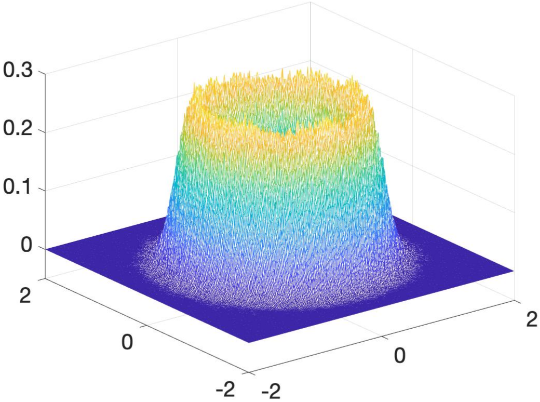
$$\rho(x, y) = \frac{1}{K} e^{-2(x^2+y^2-1)^2},$$

K is a normalizing constant.

Deterministic vector field



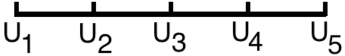
Sample density function from Monte Carlo



Solution: Data-driven PDE solver (low dimensional version)

Setting

- $i = 1, \cdots, N$ grid points. Solution vector $\mathbf{u} = (u_1, \cdots, u_N)$.
- $A\mathbf{u} = 0$: Linear relation from numerical PDE scheme.
- No boundary condition. A is not a full matrix.



$$
\begin{aligned}
(-\frac{1}{h}f_1 + \frac{\epsilon}{2h^2})u_1 - \frac{\epsilon}{h^2}u_2 + (\frac{1}{h}f_3 + \frac{\epsilon}{2h^2})u_3 &= 0 \\
(-\frac{1}{h}f_2 + \frac{\epsilon}{2h^2})u_2 - \frac{\epsilon}{h^2}u_3 + (\frac{1}{h}f_4 + \frac{\epsilon}{2h^2})u_4 &= 0 \\
(-\frac{1}{h}f_3 + \frac{\epsilon}{2h^2})u_3 - \frac{\epsilon}{h^2}u_4 + (\frac{1}{h}f_5 + \frac{\epsilon}{2h^2})u_5 &= 0
\end{aligned}
$$

Solution: Data-driven PDE solver (low dimensional version)

Use Monte Carlo data

- $\mathbf{v} = (v_1, \cdots, v_N)$ is obtained from Monte Carlo simulation.
- Use $\mathbf{v}$ as a reference for the variational problem

$$\min \|\mathbf{u} - \mathbf{v}\|^2$$
$$A\mathbf{u} = \mathbf{0}$$

Least norm solution

- Let $\mathbf{d} = -A\mathbf{v}$.
- $\mathbf{u}^* = \mathbf{v} + A^T(AA^T)^{-1}\mathbf{d}$ is called the *least norm solution* to the variational problem.

Solution: Data-driven PDE solver (low dimensional version)

- The PDE solver does not rely on the boundary condition now.
- High resolution profile for interested area.
- Still need to solve large linear system.

Mechanism

- $\mathbf{v} - \mathbf{u}^{ext}$ is a random vector.
- Optimization problem projects $\mathbf{v} - \mathbf{u}^{ext}$ to $\mathrm{Ker}(A)$.
- Projection reduces $\mathbf{u}^{*} - \mathbf{u}^{ext}$.

Error Analysis

Proposition (with M. Dobson and J. Zhai)

Consider $N \times N$ mesh. Assume entries of $\mathbf{v} - \mathbf{u}^{ext}$ be i.i.d. random variables with zero mean and variance ζ^2. Assume the PDE solver has error $O(N^{-p})$. We have

$$\mathbb{E}[\|u - u^{ext}\|_{L^2}] \leq O(N^{-1/2}\zeta) + O(N^{-p}),$$

where $\|\cdot\|_{L^2}$ is the L^2 numerical integral with respect to grid points.

Error concentration

- Empirical performance is better.
- Error concentrates at the boundary of domain.
- Most principal angles between $\mathrm{Ker}(A)$ and Θ_D are small.

Block data-driven solver

- Data-driven solver does not rely on boundary.
- Divide a large N^d domain into $(N/M)^d$ blocks with size M.
- Original cost: N^{pd}. New cost: $M^{(p-1)d}N^d$.
- Empirically M (block size) can be as small as $20-30$.
- Most error term concentrates at block boundaries.
- Very efficient for 3D and 4D problems.

Interface error and correction

- Visible interface error occurs on the interface of blocks.
- Error mainly concentrates at boundary points.
- Method 1: Small overlap ($1-4$ grids) between blocks.
- Method 2: "Half block shift" to cover all interfaces.

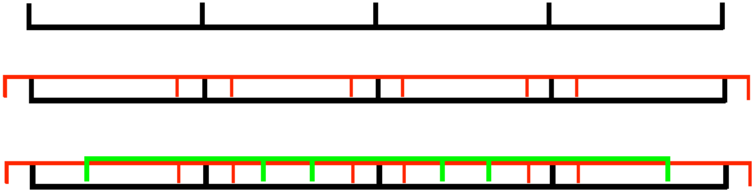
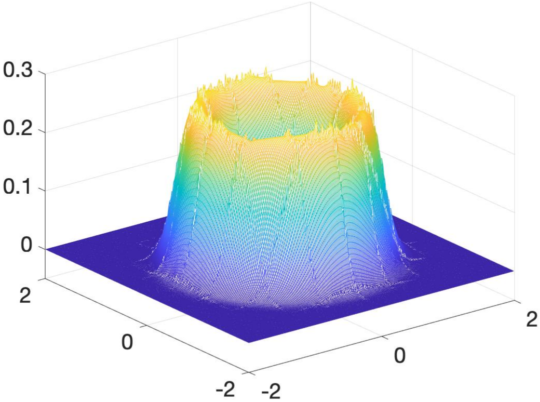
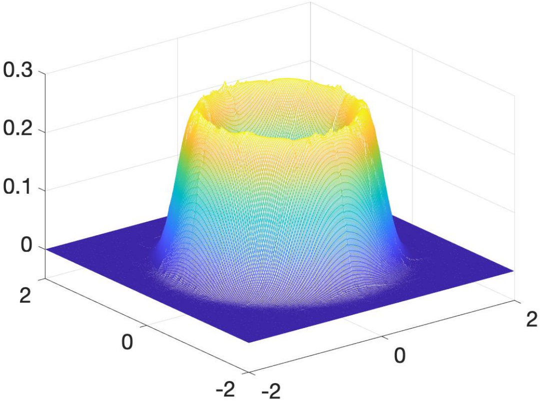


Figure: Black: blocks. Red: Overlapping numerical domain. Green: Half step shift.

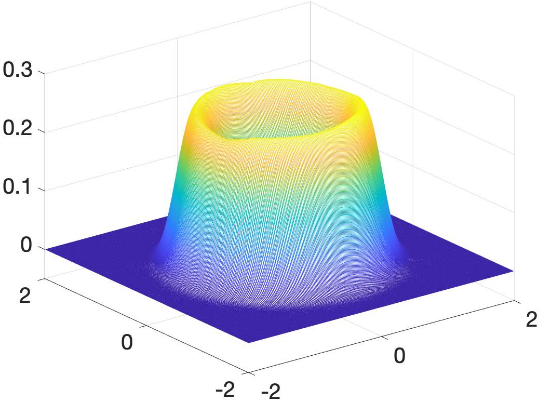
Solution without any treatment



Solution with 2-grid overlap



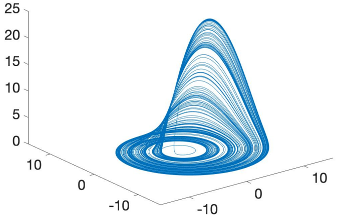
Solution after half-block shift

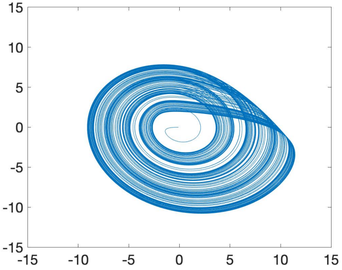


3D example: Rossler attractor

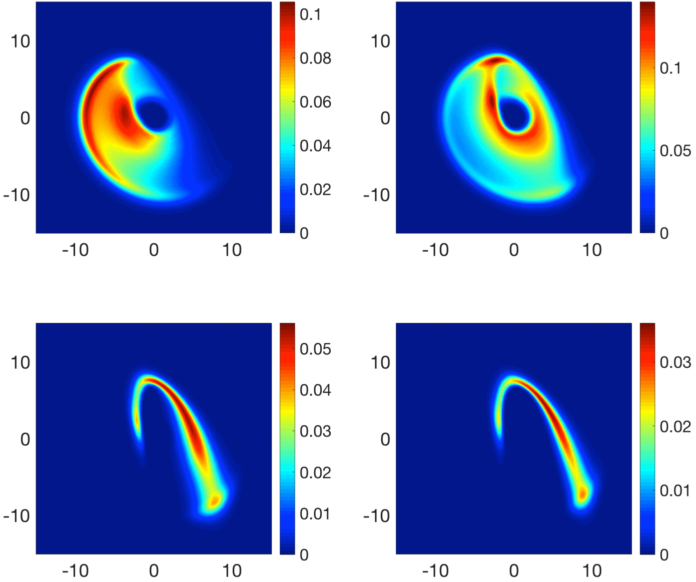
Rossler equation

$$\begin{cases} dx = (-y - z)\, dt + \varepsilon\, dW_t^x \\ dy = (x + ay)\, dt + \varepsilon\, dW_t^y \\ dz = \big(b + z(x - c)\big)\, dt + \varepsilon\, dW_t^z \end{cases},$$

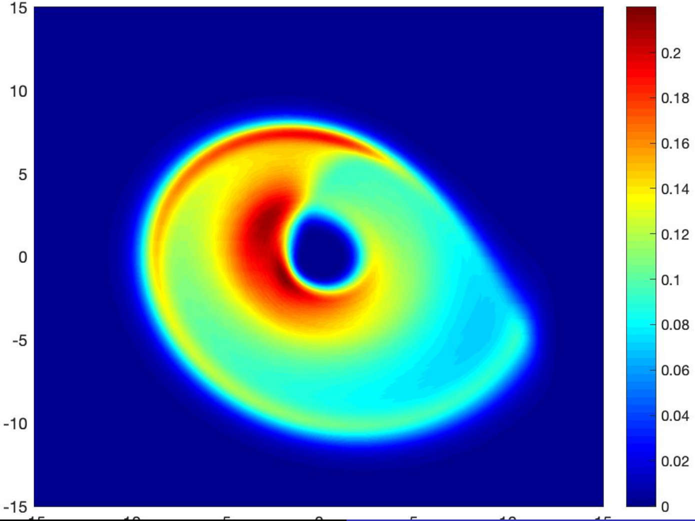




Solution on "slices"



Projection of solution



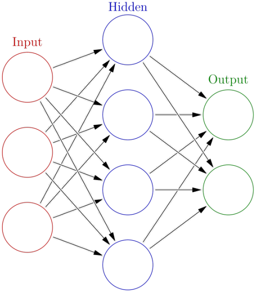
High-dimensional data-driven solver

1. Discretization does not work for high dimensional problems.
2. Higher dimension: approximate the solution by an artificial neural network $\hat{\mathbf{u}}$.
3. $\mathbf{v} = (v_1, \cdots, v_N)$ from Monte Carlo simulation.
4. New optimization problem

$$\min \|\mathbf{u} - \mathbf{v}\|^2 + \|A\mathbf{u}\|^2$$

5. What is artificial neural network?

Artificial neural network (ANN)



1. ANN is a way to approximate functions $y = NN(x, \theta)$
2. Parameter θ are coupling weights between neurons
3. Adjust θ such that $y = NN(x, \theta)$ approximates $y = f(x)$
4. Minimize a loss function $L(\theta)$ over a training set $(x_1, y_1), (x_2, y_2), \cdots$

High-dimensional data-driven solver

1. Two loss functions: $\mathcal{L}_1 = \|\hat{\mathbf{u}} - \mathbf{v}\|$, $\mathcal{L}_2 = \|\mathcal{L}^* \hat{\mathbf{u}}\|^2$.
2. Different training sets for $\mathcal{L}_1$ and $\mathcal{L}_2$.
3. Train two loss functions alternatively to avoid adjusting their weights.
4. $\mathbf{v}$ is usually a very rough approximation in high dimension. Very high spatially uncorrelated noise.

Example 1: 4D ring

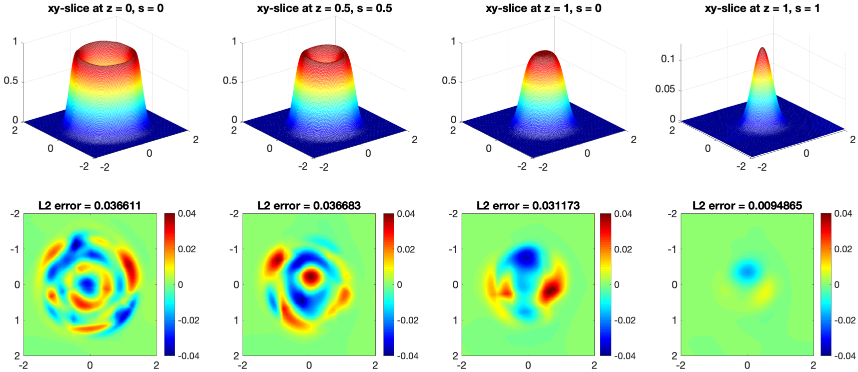
$$\begin{cases} dX_t = (-4X_t(X_t^2 + Y_t^2 + Z_t^2 + S_t^2 - 1) + Y_t)\, dt + \sigma\, dW_t^x, \\ dY_t = (-4Y_t(X_t^2 + Y_t^2 + Z_t^2 + S_t^2 - 1) - X_t)\, dt + \sigma\, dW_t^y, \\ dZ_t = (-4Z_t(X_t^2 + Y_t^2 + Z_t^2 + S_t^2 - 1))\, dt + \sigma\, dW_t^z, \\ dS_t = (-4S_t(X_t^2 + Y_t^2 + Z_t^2 + S_t^2 - 1))\, dt + \sigma\, dW_t^s, \end{cases}$$

Invariant density

$$u(x, y, z, s) = \frac{1}{K} \exp(-2(x^2 + y^2 + z^2 + s^2 - 1)^2)\,.$$

concentrate near a 4D sphere.

Example1: 4D ring



Example 2: Stochastic heat equation

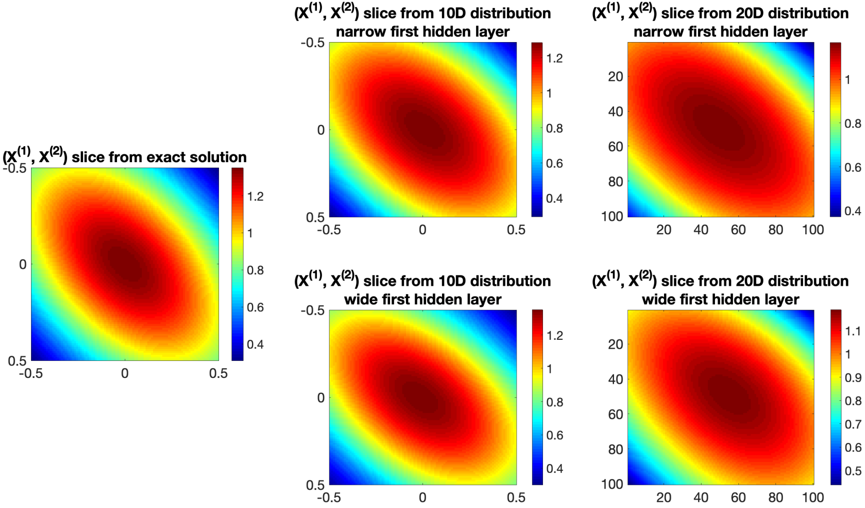
Consider a discrete stochastic heat equation

$$\mathrm{d}U_i = (U_{i-1} + U_{i+1} - 2U_i)\mathrm{d}t + \mathrm{d}W_t^{(i)}$$

for $i = 1, \cdots, N$. Assume $U_0 = U_N = 0$. Invariant probability density given by Lyapunov equation.

1. Neural network works well when $N = 10$
2. Neural network is not enough (128 hidden neurons) when $N = 20$.

Example 2: Stochastic heat equation



Power-law ergodicity

1. Find a small set C
2. Recall that power law tail of η_C is preserved in both η_α and the first simultaneous coupling.
3. Simulate the first passage time η_C
4. If $\sup_{x\in C}\mathbb{E}_x[\eta_C^\beta] < \infty$, the speed of contraction is $\sim n^{-\beta}$.
5. Use extreme value theory to verify the bounded supreme if C has high dimension.

Ref: H. Xu and Y. Li, 2017, JSP

Geometric ergodicity

Definition

X_t: Markov process with transition kernel P and invariant probability measure π.

X_t is geometrically ergodic with rate r if

$$\lim_{t\to\infty} \frac{1}{t} \log \|P^t(x,\cdot) - \pi\|_{TV} = -r$$

Importance

- r is the spectral gap for reversible X_t.
- Interplay of deterministic dynamics and noise.
- Difficult to estimate for non-gradient case. Most rigorous results are not sharp.

Recall: coupling lemma

Coupling lemma

- Let $(X_t^{(1)}, X_t^{(2)})$ be a coupling such that if $X_t^{(1)} = X_t^{(2)}$, then $X_s^{(1)} = X_s^{(2)}$ for all $s > t$.
- $\tau_C = \inf_t \{\mathcal{X}_t = \mathcal{Y}_t\}$ is the coupling time.
- $$\|\mu P^t - \nu P^t\|_{TV} \leq 2\mathbb{P}_{\mu,\nu}[\tau_C > t]\,.$$
- There exists an optimal coupling such that the equality holds.

Upper and lower bound

Lower bound

Estimate r_l such that

$$\mathbb{P}[\tau_C > t] \approx e^{-r_l t}.$$

$r_l < r$ is a lower bound of geometric ergodicity rate.

Upper bound

Construct disjoint sets (A_t, B_t). Run coupling $(\mathcal{X}_t, \mathcal{Y}_t)$ with $\mathcal{X}_0 \in A_0$ and $\mathcal{Y}_0 \in B_0$.

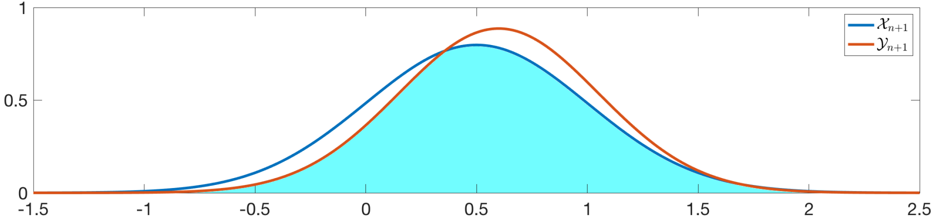
$$\xi_C = \min\left\{\inf_t\{\mathcal{X}_t \notin A_t\}, \inf_t\{\mathcal{Y}_t \notin B_t\}\right\}, \quad \mathbb{P}[\xi_C > t] \approx e^{-r_u t}$$

$r < r_u$ is an upper bound of geometric ergodicity rate.

How to couple numerically

Let $(X_t^{(1)}, X_t^{(2)})$ be a coupling.

- Independent. $X_t^{(1)}$ and $X_t^{(2)}$ are independent until coupling.
- Synchronous. Use the "same noise".
- Reflection. Use "mirroring" random terms.
- Maximal coupling. Compare density function when $|X_t^{(1)} - X_t^{(2)}| \ll 1$.



Example 1: SIR model

SIR model with degenerate noise

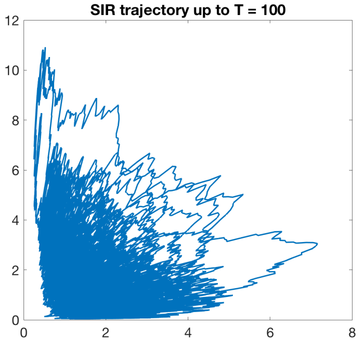
$$\mathrm{d}S = (\alpha - \beta SI - \mu S)\mathrm{d}t + \sigma S \mathrm{d}W_t$$
$$\mathrm{d}I = (\beta SI - (\mu + \rho + \gamma)I)\mathrm{d}t + \sigma I \mathrm{d}W_t \,,$$

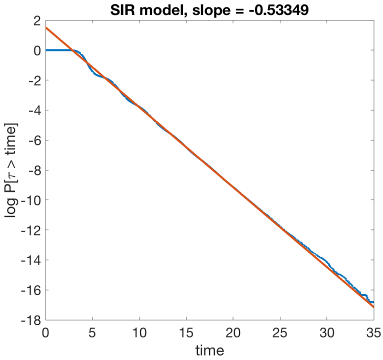
Non-degenerate invariant probability measure if $\frac{\alpha\beta}{\mu} - (\mu + \rho + \gamma - \frac{\sigma^2}{2}) > 0$. Rigorous proof only gives faster-than-power-law ergodicity (Yin et al. 2016 SIADS).

How to couple?

- Synchronous coupling until $|X_t^{(1)} - X_t^{(2)}| \ll 1$
- Compute probability density function for two steps.
- Use maximal coupling.

SIR model


SIR trajectory up to T = 100
0
2
4
6
8
10
12


SIR model, slope = -0.53349
log P[τ > time]
time
0
5
10
15
20
25
30
35
2
-18

Example 2: Coupled Fizhugh-Nagumo oscillators

50 coupled neurons.

$$\mathrm{d}u_i = \left(\frac{u}{\mu} - \frac{u^3}{3\mu} - \frac{1}{\sqrt{\mu}}v + \frac{d_u}{\mu}(u_{i+1} + u_{i-1} - 2u_i) + \frac{w}{\mu}(\bar{u} - u_i)\right)\mathrm{d}t + \frac{\sigma}{\sqrt{\mu}}\mathrm{d}W_t^1$$

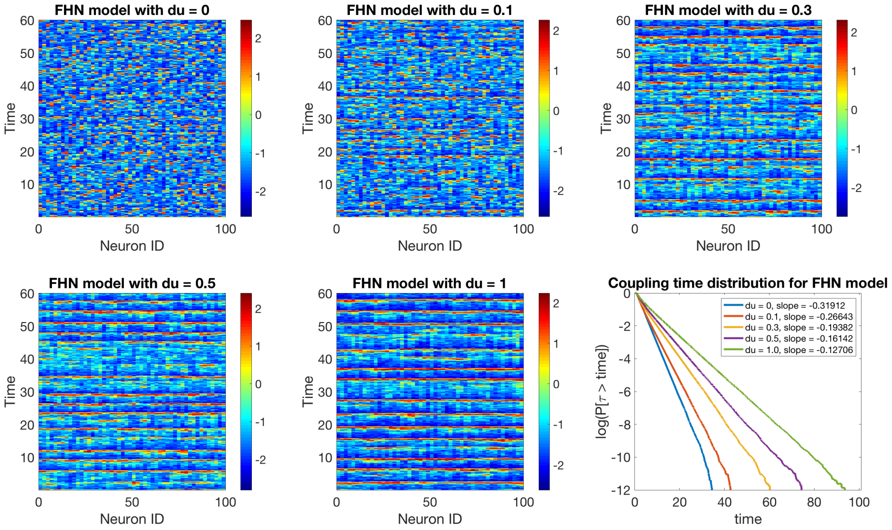
$$\mathrm{d}v_i = (\frac{1}{\sqrt{\mu}}u + \frac{a}{\sqrt{\mu}})\mathrm{d}t + \frac{\sigma}{\sqrt{\mu}}\mathrm{d}W_t^2\,.$$

- $i = 1, \cdots, 50$
- d_u: neareast-neighbor coupling strength. w: mean field coupling strength.
- $\bar{u}$: average membrane potential.

Example 2: Coupled Fizhugh-Nagumo oscillators

- $\omega = 0.4$, $\mu = 0.05$, $\sigma = 0.6$
- Change nearest-neighbor coupling strength *du*.
- Reflection coupling until $|\mathcal{X}_t - \mathcal{Y}_t| \ll 1$
- Compare probability density functions and use maximal coupling.
- Numerical result: higher *du* gives more coherent evolution, and slower rate of geometric ergodicity.
- Heuristically, strong synchronization makes two trajectories harder to couple.

Coupled Fizhugh-Nagumo oscillators



Comments

- Coupling method is data-driven. No spatial discretization.
- Coupling speed versus noise magnitude depends on the deterministic dynamics.
- Important application: classification of high dimensional potential landscape.
- Ongoing work: *Can you hear the shape of a landscape?*

Selected Reference

- Li, Y. *A data-driven method for the steady state of randomly perturbed dynamics*, Communications in Mathematical Sciences, 17(4), 1045-1059, 2019

- Dobson, M., Li, Y. and Zhai, J. *An efficient data-driven solver for Fokker-Planck equations: algorithm and analysis*, Communications in Mathematical Sciences, accepted

- Dobson, M., Li, Y. and Zhai, J. *A deep learning method for solving Fokker-Planck equations* , MSML21: Mathematical and Scientific Machine Learning, Aug 16-19, 2021

- Li, Y., Wang, S. *Numerical computations of geometric ergodicity for stochastic dynamics* , Nonlinearity 33(12), 6935, 2020

- Li, Y., Xu, H. *Numerical Simulation of Polynomial-Speed Convergence Phenomenon* Journal of Statistical Physics, 169(4), 2017

Thank you