Density Tracking by Quadrature for SDE Inference

Harish S. Bhat*, R. W. M. A. Madushani*, Shagun Rawat*

*Applied Mathematics Unit, University of California, Merced

Motivation (starting from the data)

- Let's say you have many time series observations.
    - Perhaps the observations are at non-equispaced, irregular times.
    - Perhaps there are multiple time series, i.e., independent observations of the same process.
- *How can we use this data to infer both predictive and explanatory SDE models?*

Motivation (starting from the model)

- Stochastic differential equations (SDE) are widely used to model time-dependent phenomena.
  - Such models often have coefficients or parameters that must be determined from data.
  - Typically we only have noisy, imprecise observations of the states.
  - We seek methods for jointly inferring states and parameters in SDE models.

Motivation (even more!)

- For most SDE of interest, the likelihood function cannot be computed analytically.
    - How can we efficiently compute a convergent approximation to the likelihood function?
    - How do we incorporate this computation into a Metropolis algorithm?

Stochastic Differential Equation (SDE) Notation:

- We consider models of the form:

$$dX(t) = f(X(t), \theta)dt + g(X(t), \theta)dW_t$$

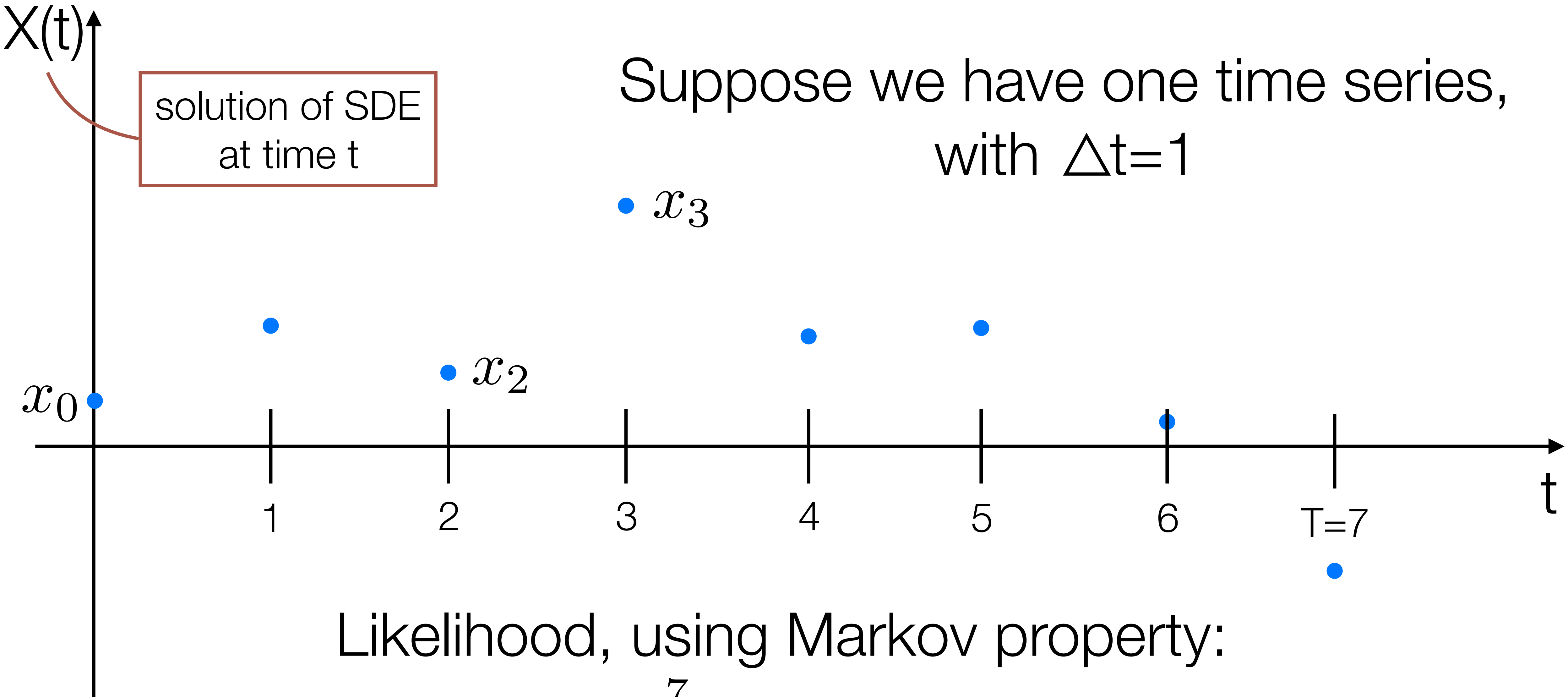
- $X(t)$ is the solution of the SDE at time t.
- W_t is Brownian motion (Wiener process).
- θ is a vector of parameters, $\mathbf{x}$ is the data
- Goal: sample from posterior $p(\theta|\mathbf{x})$

Bayes

$$p(\theta|\mathbf{x}) = \frac{p(\mathbf{x}|\theta)p(\theta)}{p(\mathbf{x})}$$

$$\text{posterior} = \frac{\text{likelihood} \cdot \text{prior}}{\text{normalization constant}}$$

Pictorial Representation (why is the problem hard?)



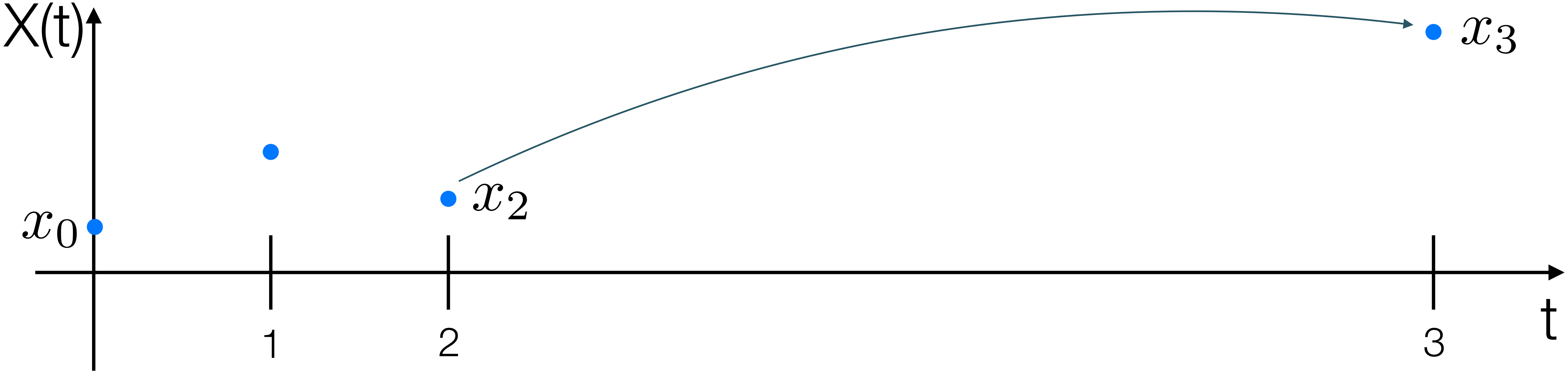
Suppose we have one time series, with $\triangle$t=1

Likelihood, using Markov property:

$$p(\mathbf{x}|\theta) = p(X(0) = x_0|\theta) \prod_{j=1}^{7} p(X(j) = x_j | X(j-1) = x_{j-1}, \theta)$$

Pictorial Representation

Let's zoom in on 2 ≤ t ≤ 3.



Piece of the likelihood: $\underbrace{p(X(3) = x_3 | X(2) = x_2, \theta)}_{\text{transition density: no analytical formula (for most SDE)}}$

Context

There is a large literature
on inference for SDEs.

Two main strategies:

Deterministic

Stochastic

Context

- Large literature on Bayesian inference for SDEs, with two main lines of attack:

    1. **Deterministic**: use series expansions to analytically approximate transition density—Iacus [2008, 2014]. Or, use Fokker-Planck/Kolmogorov PDE.

    2. **Stochastic**: construct sample paths at intermediate time points, using concepts like the Brownian bridge, and numerically evaluate transition density. See Fuchs [2013].

- Our approach is deterministic and numerical.

Comparison

- Hurn, Jeisman, and Lindsay [2007] compared many methods for SDE inference and found:
  - Solving the Fokker-Planck PDE to compute transition densities yields most accurate inference.
  - The only drawback is speed.

Density Tracking by Quadrature (DTQ)

- This is **how** we step forward in time.
- Start with the SDE:

$$dX(t) = f(X(t), \theta)dt + g(X(t), \theta)dW_t$$

- Euler-Maruyama approximation of SDE:

independent standard normal

$$X(t_{i+1}) = X(t_i) + f(X(t_i))h + g(X(t_i))h^{1/2}Z_{i+1}$$

$t_{i+1} - t_i$

- This approximation implies:

$$X(t_{i+1})|X(t_i) = y \ \sim \ \mathcal{N}(\mu = y + f(y)h, \sigma^2 = g^2(y)h)$$

Density Tracking by Quadrature (DTQ)

$$X(t_{i+1}) | X(t_i) = y \;\sim\; \mathcal{N}(\mu = y + f(y)h, \sigma^2 = g^2(y)h)$$
$$X(t_{i+1}) = X(t_i) + f(X(t_i))h + g(X(t_i))h^{1/2} Z_{i+1}$$

- Above two equations imply (Chapman-Kolmogorov)

$$p(x, t_{i+1}) = \int_{\mathbb{R}} p_{X(t_{i+1})|X(t_i)=y}(x)\, p(y, t_i)\, dy$$

- If computers could compute in continuous space, this would be a method to step the PDF forward in time.

Density Tracking by Quadrature (DTQ): Missing Pieces

- Fix a spatial grid $x_m = y_m = m\Delta x$
- Represent $p(y, t_i)$ as a finite-dimensional vector

$$\mathbf{p}_i = \{p(y_m, t_i)\}_{m=-M}^{m=M}$$

- Truncate integral, apply trapezoidal rule to Chapman-Kolmogorov:

$$p(x, t_{i+1}) = \int_{\mathbb{R}} p_{X(t_{i+1})|X(t_i)=y}(x)\, p(y, t_i)\, dy$$

- Whole thing reduces to iterated matrix multiplication!

$$\mathbf{p}_{i+1} = A\mathbf{p}_i$$

DTQ Preprint and Code

- For more information on DTQ itself, consult:
  - Bhat and Madushani [2016],
    Density tracking by quadrature for stochastic differential equations, arXiv:1610.09572.
- To try DTQ, see the R package on CRAN:
  - Rdtq (https://cran.r-project.org/package=Rdtq)
- For the source code, see:
  - https://github.com/hbhat4000/Rdtq

DTQ Theoretical Results

- $p(x, t)$ exact PDF of the SDE
- $\tilde{p}(x, t)$ exact PDF of the Euler-Maruyama approximation
- $\hat{p}(x, t)$ what DTQ computes

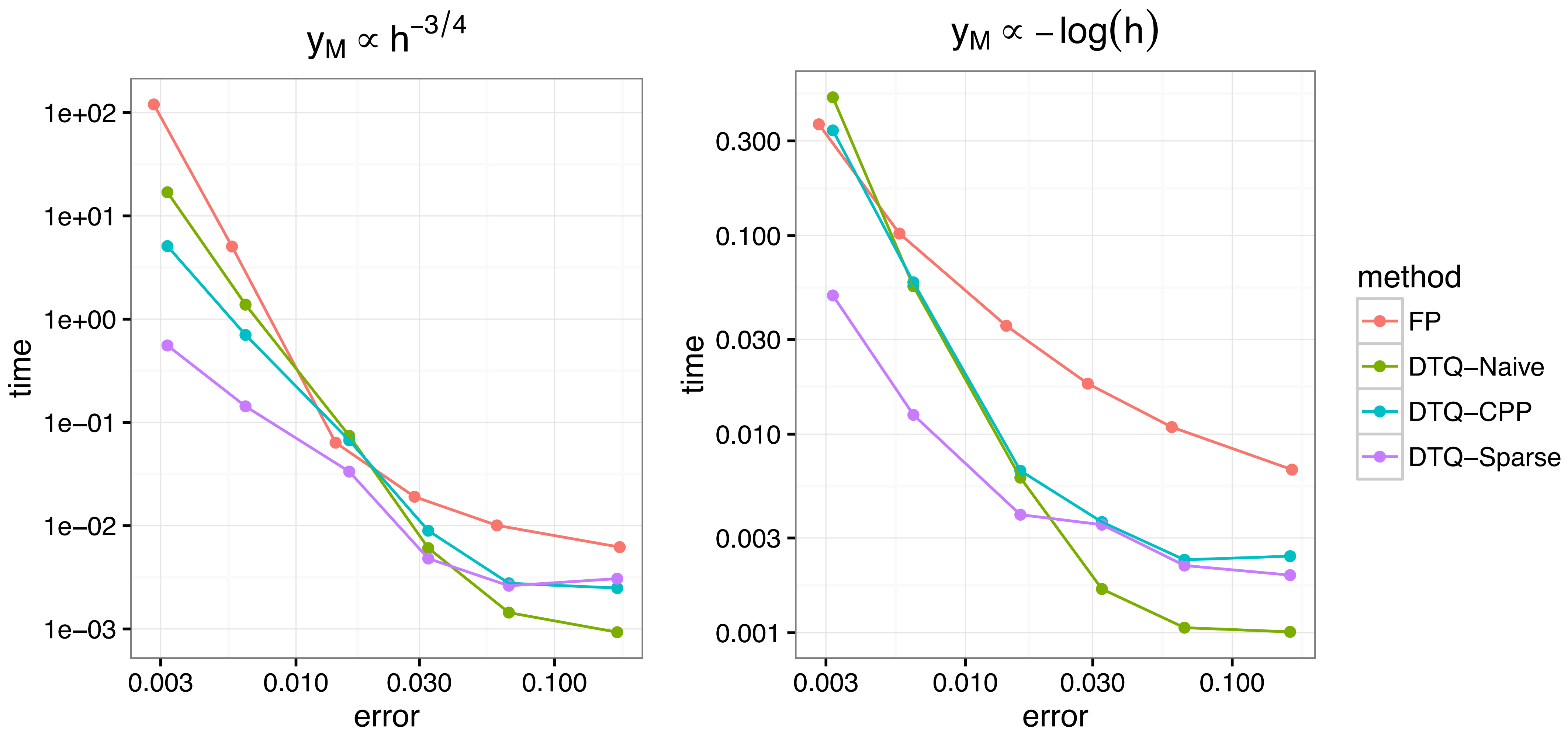
We have proved

$$\|\hat{p}(\cdot, T) - \tilde{p}(\cdot, T)\|_{L^1} = O(h^{-1} \exp(-r h^{-\kappa}))$$

Bally and Talay [1996] proved

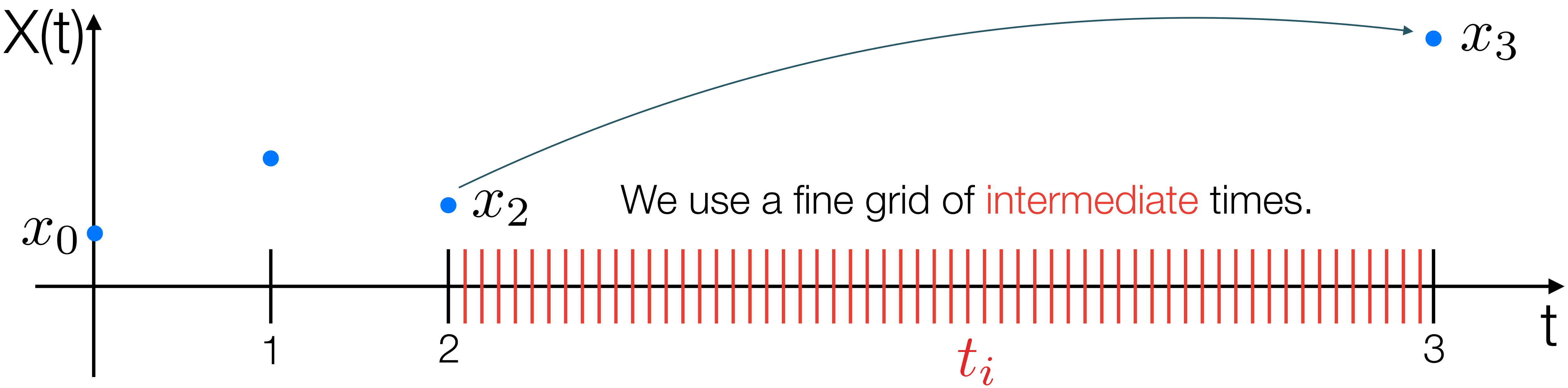
$$\|\tilde{p}(\cdot, T) - p(\cdot, T)\|_{L^1} = O(h)$$

DTQ Numerical Comparison



At the finest error level, DTQ-Sparse is 10-100x faster than Fokker-Planck

Our Approach (One Sample Path)



How do we think about $p(X(3) = x_3 | X(2) = x_2, \theta)$?

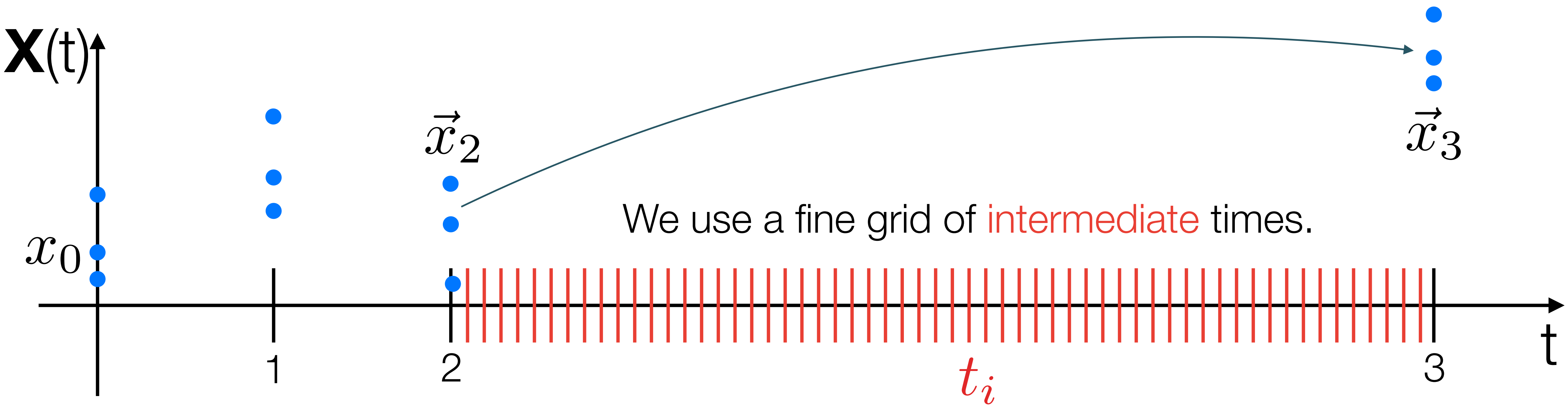
Let $p(x, t)$ denote the p.d.f. of $X(t)$, for fixed θ

Start with $p(x, 2) = \delta(x - x_2)$

Step forward in time to solve for $p(x, t_i)$

Evaluate $p(x_3, 3) \approx p(X(3) = x_3 | X(2) = x_2, \theta)$

Our Approach (Many Sample Paths)



When we have M sample paths, only one change:

Start with $p(x,2) = \frac{1}{M}\sum_{m=1}^{M}\delta(x - x_2^m)$

Step forward in time to solve for $p(x, t_i)$

Evaluate $\prod_{m=1}^{M} p(x_3^m, 3) \approx p(\vec{X}(3) = \vec{x}_3 | \vec{X}(2) = \vec{x}_2, \theta)$

Metropolis Algorithm

- Start with initial $\vec{\theta}^{(i)}$
- Proposal: $\vec{\theta}^{*} = \vec{\theta}^{(i)} + \vec{Z}$
- Ratio: $\rho = \dfrac{p\left(\vec{\theta}^{*}|\mathbf{x}\right)}{p\left(\vec{\theta}^{(i)}|\mathbf{x}\right)} = \dfrac{p\left(\mathbf{x}|\vec{\theta}^{*}\right)p(\vec{\theta}^{*})}{p\left(\mathbf{x}|\vec{\theta}^{(i)}\right)p(\vec{\theta}^{(i)})}$

Likelihoods computed via DTQ method

- Let $u \sim U(0,1)$. Accept if $\rho > u$; then $\vec{\theta}^{(i+1)} = \vec{\theta}^{*}$
- Else reject; then $\vec{\theta}^{(i+1)} = \vec{\theta}^{(i)}$

Example

- Consider nonlinear SDE:

$$dX(t) = \theta_1 X(t) \left(\theta_2 - X(t)^2\right) dt + e^{\theta_3} dW_t$$

- We generate simulated data using

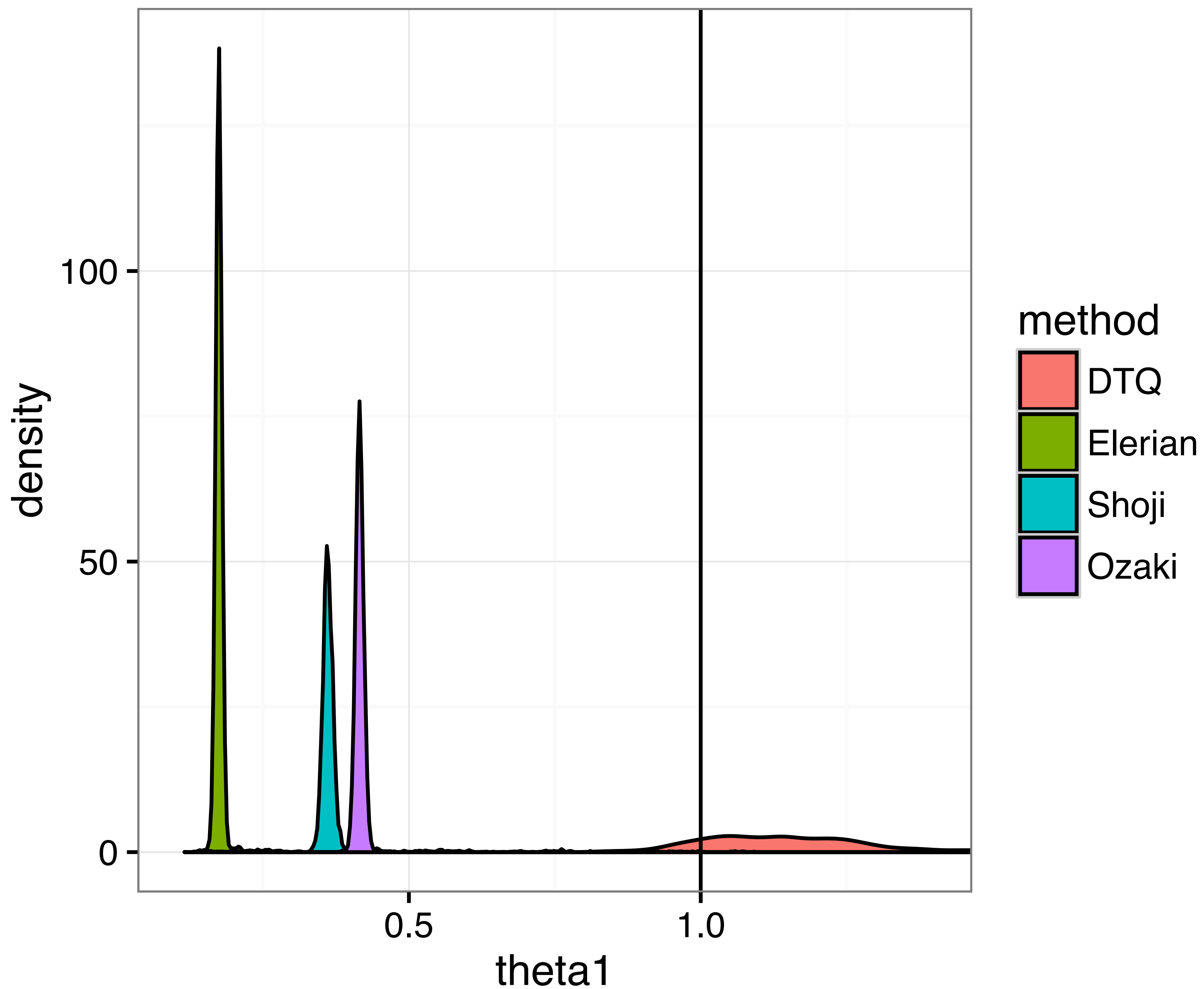
$$\theta_1 = 1,\ \theta_2 = 4,\ e^{\theta_3} = 0.5$$

- Simulation parameters:
  - 100 sample paths from t=0 to T=25.
  - Euler-Maruyama method with internal time step of h=0.0001.
  - However, data is only recorded at times 0, 1, 2, …, 25.

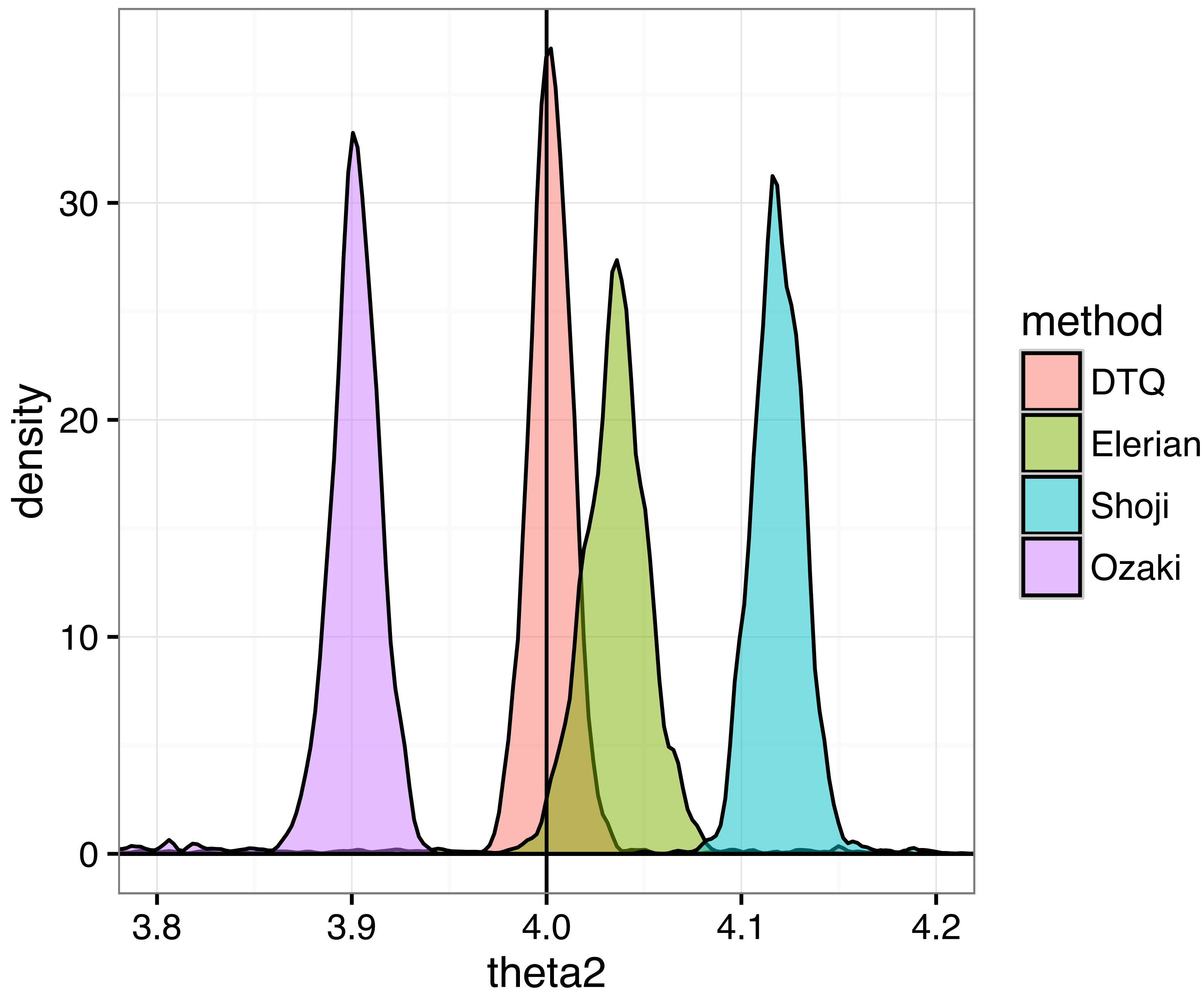
Inference Test

- Consider 4 competing methods to compute likelihood:
  - Kessler, Ozaki, Shoji, Elerian (deterministic methods, CRAN “sde” package)
- Normal prior with $\mu = 0.5, 0.5, 0;\ \ \sigma = 4$
  - not very close to ground truth
- Normal proposal with $\mu = 0;\ \ \sigma = 0.02, 0.02, 0.01$
  - Acceptance rates between 20.5% and 43%
- Initialize Metropolis at MLE: $\vec{\theta}^{(0)} = (0.925, 3.99, 0.43)$
  - Generate 10000 samples of posterior, discard first 100. No thinning.

Results: [1/3]

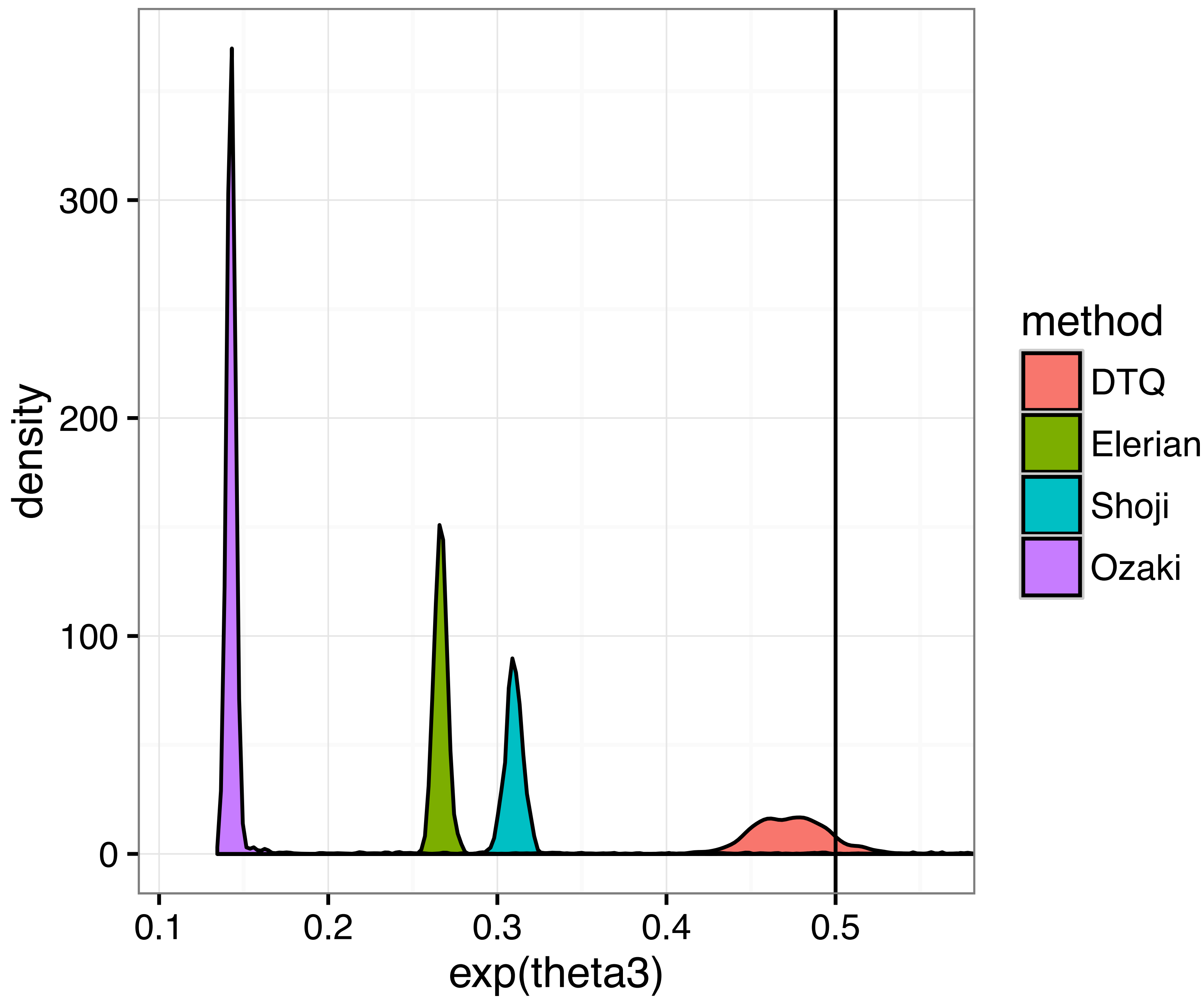


Results: [2/3]



density
30
20
10
0
3.8
3.9
4.0
4.1
4.2
theta2
method
DTQ
Elerian
Shoji
Ozaki

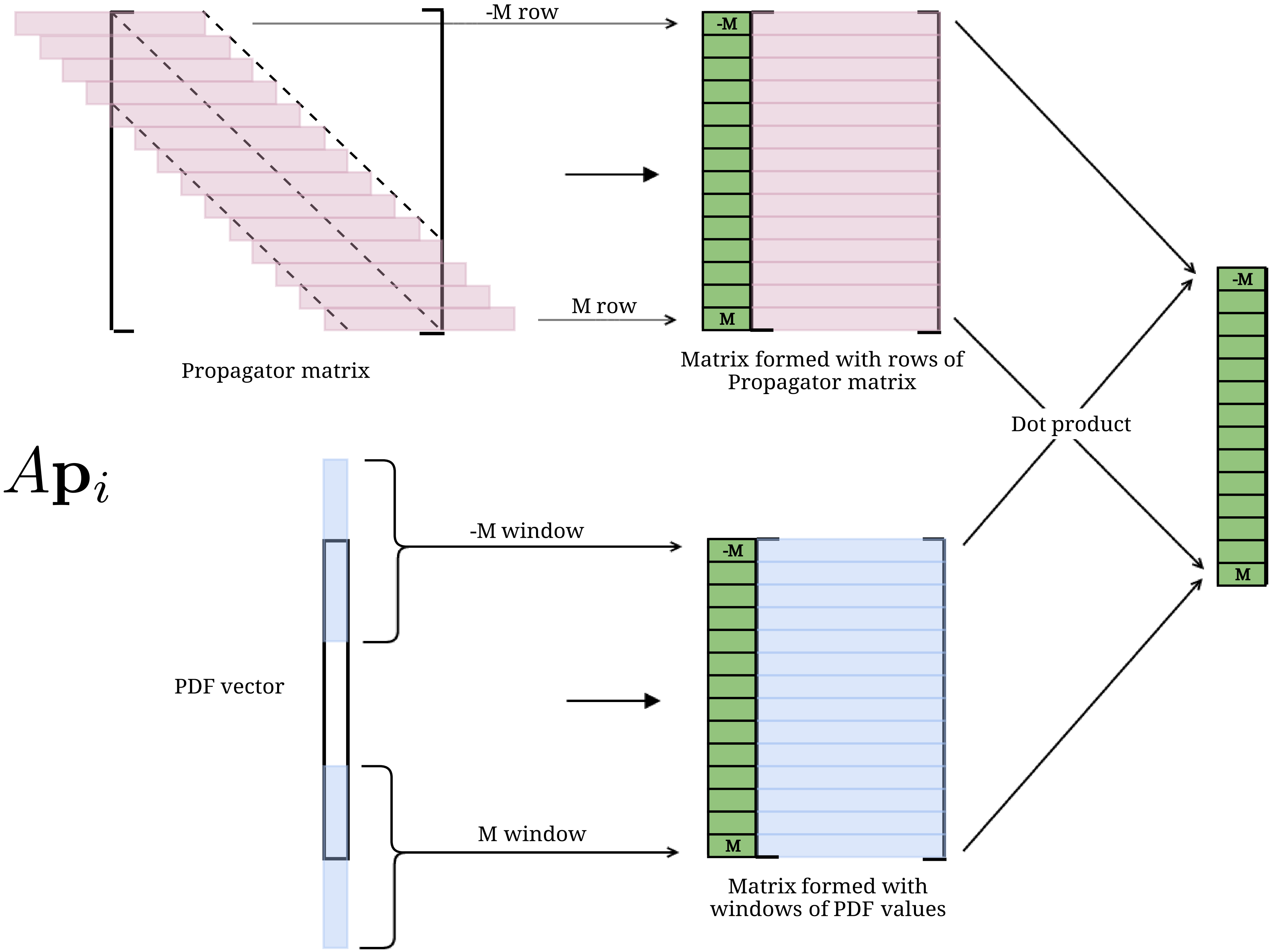
Results: [3/3]



300
200
100
0
density
0.1
0.2
0.3
0.4
0.5
exp(theta3)
method
DTQ
Elerian
Shoji
Ozaki

Natural Parallelization: Part I (Scala, Breeze, MKL)

$$\mathbf{p}_{i+1} = A\mathbf{p}_i$$



Natural Parallelization: Part II (Spark)

$$X(t_{i+1})|X(t_i) = y \; \sim \; \mathcal{N}(\mu = y + f(y)h, \sigma^2 = g^2(y)h)$$

implies that the transition kernel is time-independent.

$$A_{ab} = p_{X(t_{i+1})|X(t_i)=y_b}(x_a)$$

Therefore, can compute in parallel **all** terms

$$p(\vec{X}(j) = \vec{x}_j | \vec{X}(j-1) = \vec{x}_{j-1}, \theta)$$

We do this in Spark using **`sc.parallelize`** and **`map`**.

Extensions: Filtering and Inference

- Consider model:

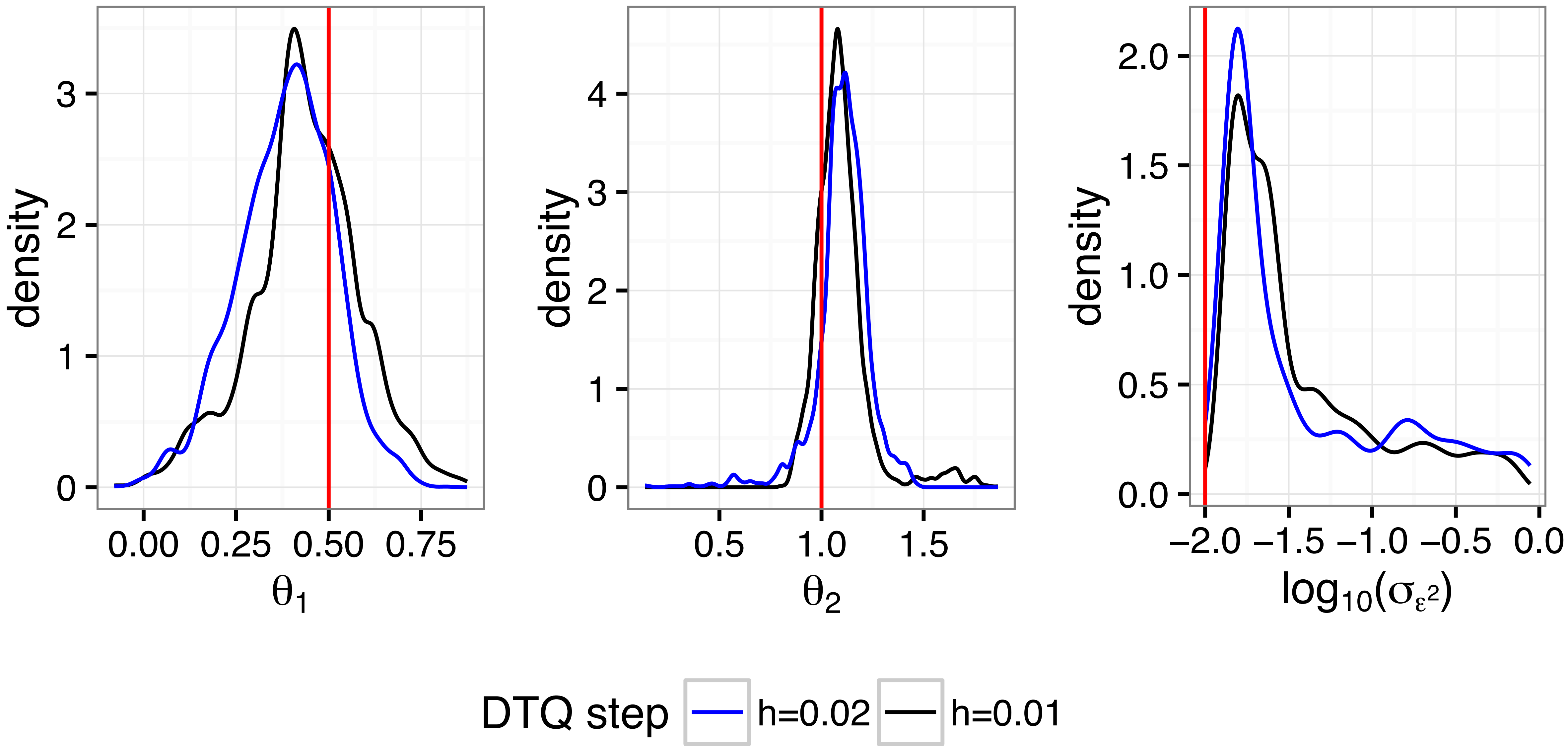
$$dX(t) = f(X(t), \theta)dt + g(X(t), \theta)dW_t$$

$$Y(t) = X(t) + \epsilon_t \qquad \epsilon_t \sim \mathcal{N}(\mu = 0, \sigma^2 = \sigma_\epsilon^2)$$

- Y(t) = observations = SDE solution (or state) + noise.
- Data $\mathbf{y}$ = Y(t), sampled at irregular times.
- Goals: infer both parameters and states.
- Sample from joint posterior $p(\mathbf{x}, \theta, \sigma_\epsilon^2 | \mathbf{y})$

For more details, see our KDD BigMine '16 paper.

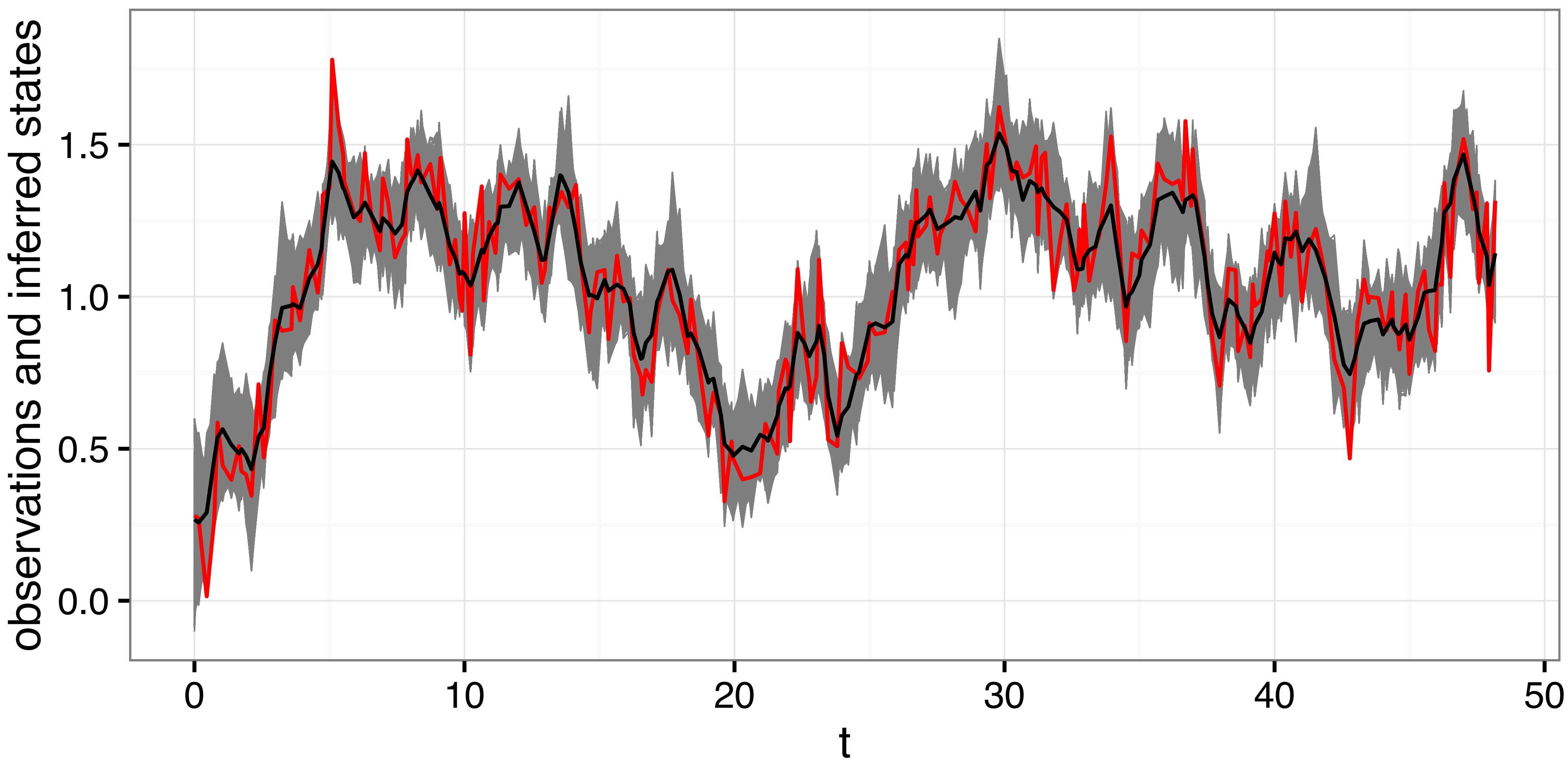
Results: Posterior Densities of Parameters



$$dX(t) = \theta_1(\theta_2 - X(t))dt + 0.25dW_t$$

$$Y(t) = X(t) + \epsilon_t$$

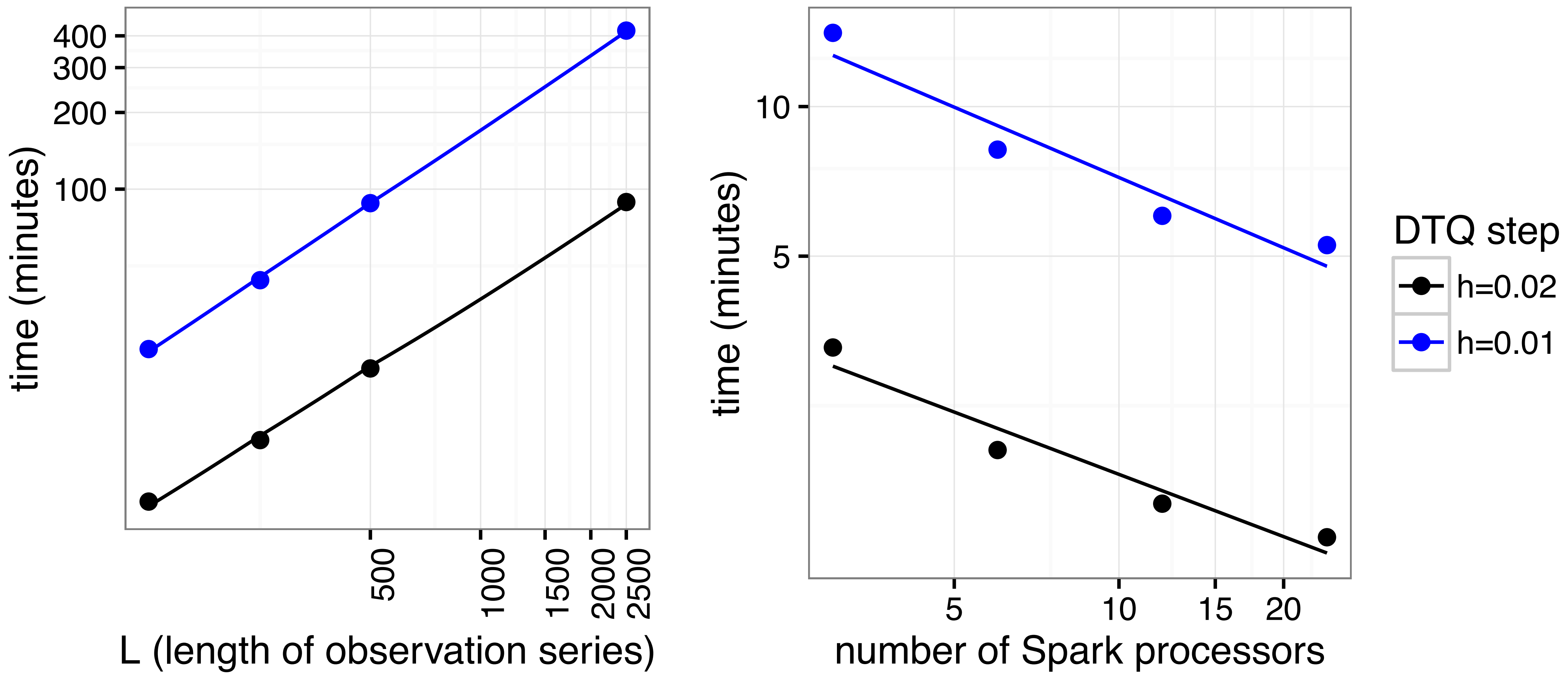
Results: Inference of States X(t) from Observations Y(t)



$$dX(t) = \theta_1(\theta_2 - X(t))dt + 0.25dW_t$$

$$Y(t) = X(t) + \epsilon_t$$

Results: Scaling



Nonparametric Inference

- Hermite functions form orthonormal basis for L^2:

$$\psi_j(x) = (-1)^j (2^j j! \sqrt{\pi})^{-1/2} e^{x^2/2} \frac{d^j}{dx^j} e^{-x^2}$$

- We write our unknown functions as linear combinations of these basis functions, i.e.,

$$f(x) \approx \sum_{i=0}^{N_f} \theta_i \psi_i(x) = \hat{f}(x; \boldsymbol{\theta})$$

$$g(x) \approx \sum_{i=0}^{N_g} \theta_{N_f+1+i} \psi_i(x) = \hat{g}(x; \boldsymbol{\theta})$$

- Then the problem is to find the parameter vector $\boldsymbol{\theta}$

Adjoint Method: Problem

- In nonparametric inference, # of parameters is

$$N_f + N_g + 2$$

- To compute gradient of likelihood w.r.t. parameters via "direct method," we take $\frac{d}{d\theta_i}$ of the DTQ equation

$$p(x, t_{i+1}) = \int_{-y_M}^{y_M} p_{X(t_{i+1})|X(t_i)=y}(x) p(y, t_i) \, dy$$

- This will give us one evolution equation per parameter.
- We have tried this: *resulting optimization of log likelihood is too slow to be practical*.

Adjoint Method: Solution

- How do we compute gradient of likelihood w.r.t. parameters via the adjoint method?

- First, introduce u, a variable that is adjoint or dual to p

- Then derive from DTQ an evolution equation for u

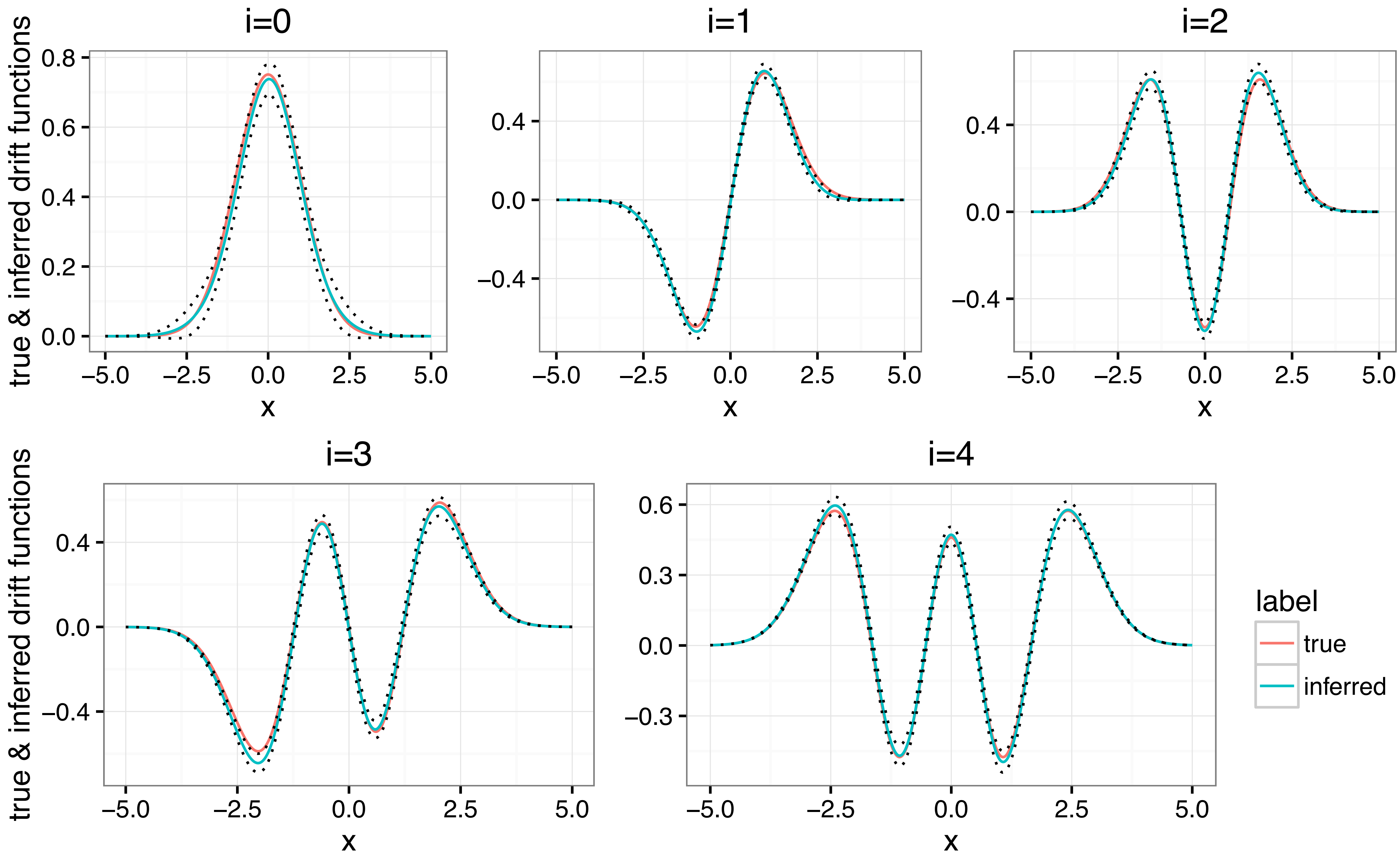
- This evolution equation proceeds backwards in time. If we solve it **once**, we get the entire gradient.

- This huge cost-savings is the key technical innovation of our work that enables practical inference.

First Set of Results

- Consider SDE with constant diffusion $g(x) \equiv 1$ and drift function equal to a Hermite basis function,
$$f(x) = \psi_i(x)$$
- We simulate 10000 sample paths of this SDE from t=0 to t=4, using a small internal time step of 10^{-4}.
- Solution is retained only at t=0,1,2,3,4.
- We then take $N_f = 4$ and proceed with inference.
- Ground truth: for $f(x) = \psi_i(x)$, $\theta_j = \delta_{j,i}$
$$g(x) \equiv 1 \qquad \theta_5 = 1$$
- Initialize trust region optimizer with $\boldsymbol{\theta} = (1, 1, \ldots, 1)$

First Set of Results



Model Selection/Regularization

- Two main approaches:
  1. Find the best # of basis functions for f and g
  2. *Choose a really large # of basis functions; regularize using a penalty term*

$$E = \int_{x=-\infty}^{\infty} \left| \hat{f}'(x) \right|^2 \, dx$$

$$J(\boldsymbol{\theta}) = -\log \mathcal{L}(\boldsymbol{\theta}) + \gamma E(\boldsymbol{\theta})$$

Quadratic penalty, similar to ridge regression.

For Hermite basis, can evaluate penalty easily.

Can use cross-validation to select γ.

Other Results (ask me later)

1. Replace Euler-Maruyama with higher-order method to obtain overall second-order convergence
2. Levy SDE; track characteristic fn instead of density
3. Online inference
4. Details of fast adjoint method to compute gradient of log likelihood w.r.t. theta
5. Expectation maximization
6. Higher-dimensional version + spatial tracking data

Thank You!

Code and Papers

All of our code is open source:

https://github.com/hbhat4000/sdeinference/

Papers available here:

http://faculty.ucmerced.edu/hbhat/publications.html

Email: hbhat@ucmerced.edu