Atlas Simulation: A Numerical Scheme for Multiscale Diffusions

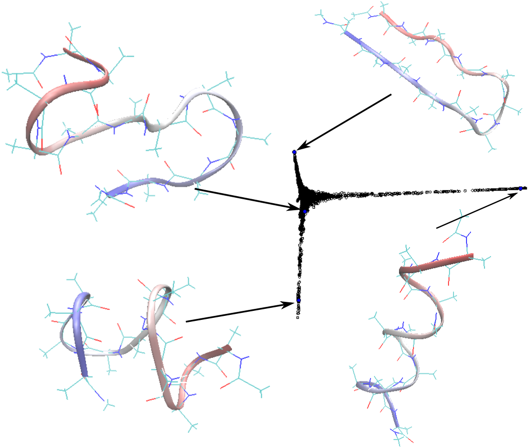
Miles Crosskey
Duke University

March 27, 2014

Advisor: Mauro Maggioni

Motivation: Molecular Dynamics

- High dimensional state space
- Low intrinsic dimension
- Small time steps
- Non-Euclidean distance



Problem Statement

Given:

- stochastic simulator on/near a manifold $\mathcal{M}$
- distance function
- homogenization scale δ

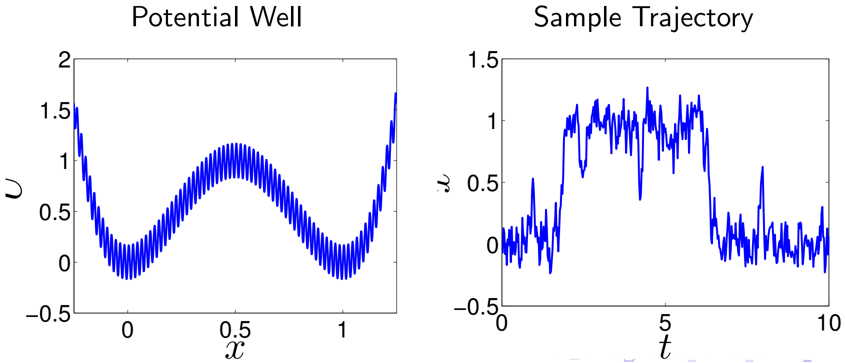
Goals:

- fast simulation of long paths
- efficient storage of long paths

1-d Example

Brownian motion in potential well

$$\dot{X}_t = -\nabla U(X_t) + \dot{W}_t$$

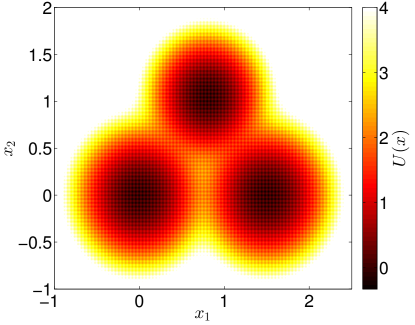


2d Example

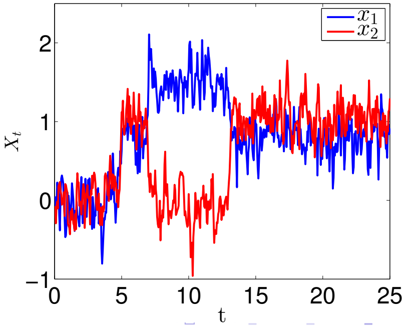
Brownian motion in 2-d potential well

$$\dot{X}_t = -\nabla U(X_t) + \dot{W}_t$$

Potential Well



Sample Trajectory



2-d $\rightarrow$ 12,500-d

Map 2-d vectors to images using the following procedure:

- Start with 2d vector x
- Map each pixel in 100 x 125 image to grid location
- Turn on pixel if grid point within $1/2$ of x

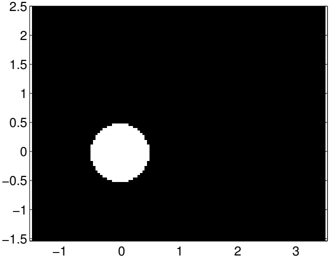


Figure : Image returned from [0,0]

12,500-d Example

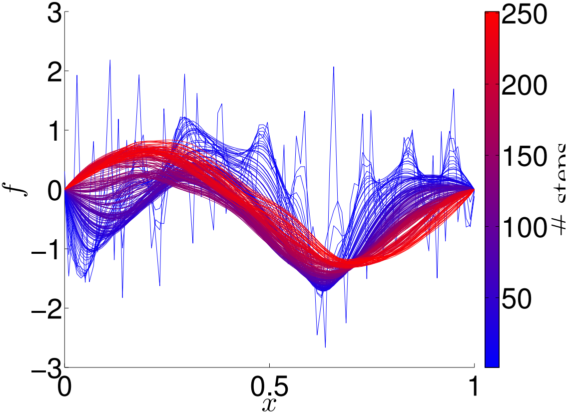
Long simulation:

- each step: run 2-d model, map to image
- each state: 100 x 125 pixel image
- video: $75,000$ steps

100-d Example

At each step starting at f:

1. Gen. Brownian bridge B
2. $f = f + (1/100) * B$
3. $f = \text{smooth}(\ f\ )$
4. $f = f \times ||f_0||/||f||$



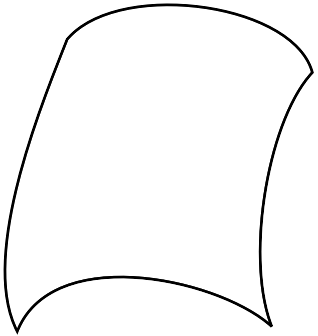
100-d Example (cont'd)

Long simulation:

- video: $30,000$ steps
- two stable states: up, down
- rare transitions

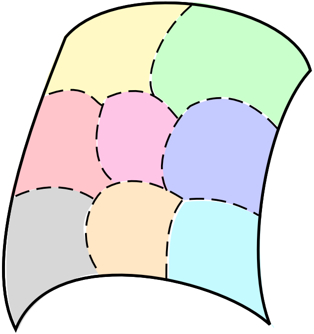
Learning the Atlas

- Divide configuration space
- Fit charts to pieces
- Connect neighboring charts
- Learn simulators on charts



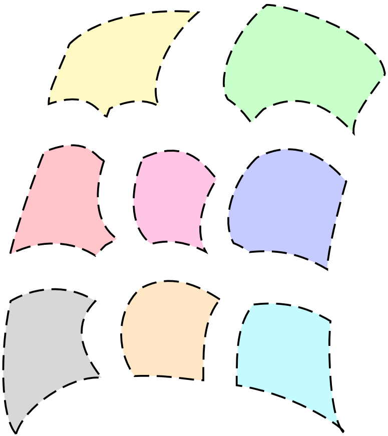
Learning the Atlas

- Divide configuration space
- Fit charts to pieces
- Connect neighboring charts
- Learn simulators on charts



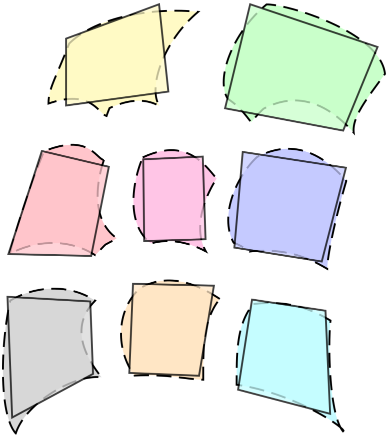
Learning the Atlas

- Divide configuration space
- Fit charts to pieces
- Connect neighboring charts
- Learn simulators on charts



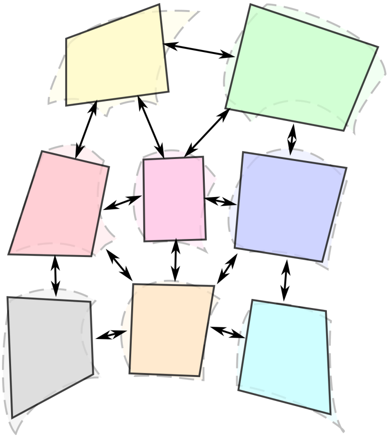
Learning the Atlas

- Divide configuration space
- Fit charts to pieces
- Connect neighboring charts
- Learn simulators on charts



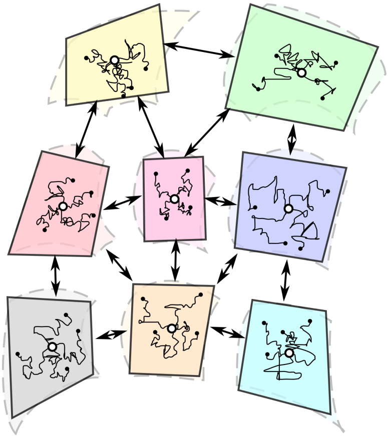
Learning the Atlas

- Divide configuration space
- Fit charts to pieces
- Connect neighboring charts
- Learn simulators on charts



Learning the Atlas

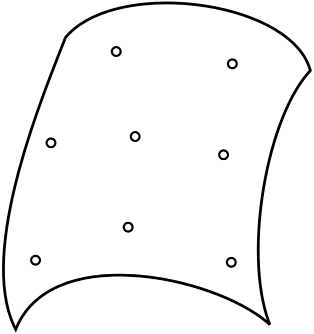
- Divide configuration space
- Fit charts to pieces
- Connect neighboring charts
- Learn simulators on charts



Dividing the space: δ-net

Given δ, find $\{y_k\}$ such that:

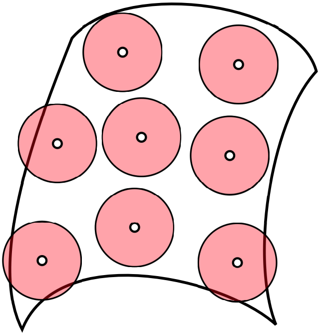
- $\delta/2$ balls around $\{y_k\}$ do not intersect.
- δ balls around $\{y_k\}$ cover the state space.
- y_k neighbors y_j if $|y_k - y_j| < 2\delta$.



Dividing the space: δ-net

Given δ, find $\{y_k\}$ such that:

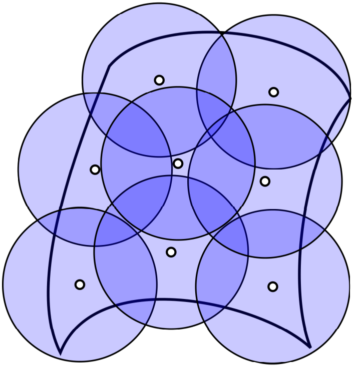
- $\delta/2$ balls around $\{y_k\}$ do not intersect.
- δ balls around $\{y_k\}$ cover the state space.
- y_k neighbors y_j if $|y_k - y_j| < 2\delta$.



Dividing the space: δ-net

Given δ, find $\{y_k\}$ such that:

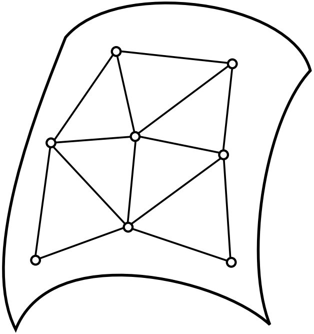
- $\delta/2$ balls around $\{y_k\}$ do not intersect.
- δ balls around $\{y_k\}$ cover the state space.
- y_k neighbors y_j if $|y_k - y_j| < 2\delta$.



Dividing the space: δ-net

Given δ, find $\{y_k\}$ such that:

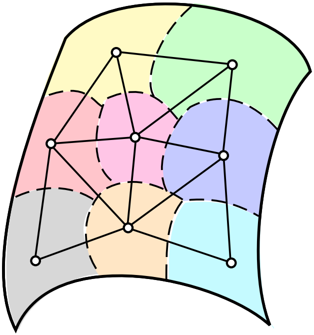
- $\delta/2$ balls around $\{y_k\}$ do not intersect.
- δ balls around $\{y_k\}$ cover the state space.
- y_k neighbors y_j if $|y_k - y_j| < 2\delta$.



Dividing the space: δ-net

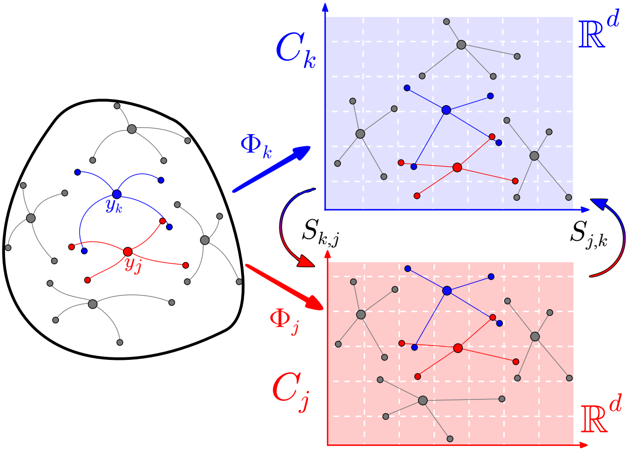
Given δ, find $\{y_k\}$ such that:

- $\delta/2$ balls around $\{y_k\}$ do not intersect.
- δ balls around $\{y_k\}$ cover the state space.
- y_k neighbors y_j if $|y_k - y_j| < 2\delta$.



Atlas Construction

- Generate landmarks
- Embed with neighbors to $\mathbb{R}^d$
- Learn linear switching map



Landmark Multi-dimensional Scaling (LMDS)

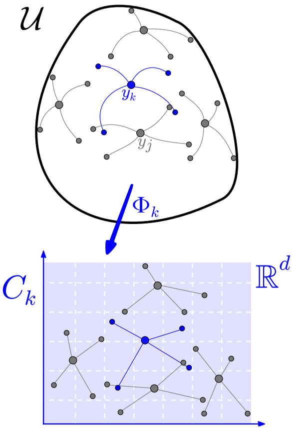
Given a neighborhood $\mathcal{U}$ with distant ρ and a set of landmarks $\{x_i\}$:

- LMDS computes $\Phi : \{x_i\} \to \mathbb{R}^d$ minimizing

$$\sum_i \sum_j (\rho(x_i, x_j)^2 - |\Phi(x_i) - \Phi(x_j)|^2)^2$$

- Extend $\Phi : M \to \mathbb{R}^d$ to $z \in U$ minimizing

$$\sum_i (\rho(x_i, z)^2 - |\Phi(x_i) - \Phi(z)|^2)^2$$



Transition maps between charts

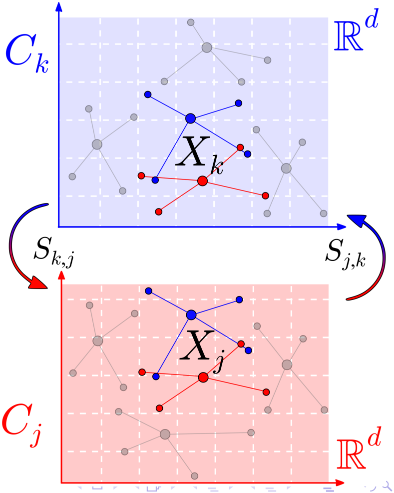
Given $n \times d$ mean zero matrices X_k, X_j:

$$S = X_k^\dagger X_j$$

minimizes

$$||X_k S_{k,j} - X_j||_2$$

over all possible $d \times d$ matrices $S_{k,j}$.

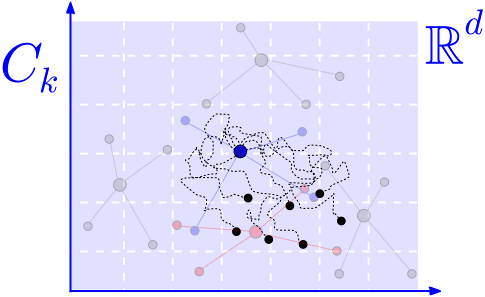


Parameter Estimation

On chart C_k associated with y_k, we fit a constant coefficient SDE

$$d\bar{X}_t = \bar{b}dt + \bar{\sigma}dB_t$$

- Pick t_0 such that X_t moves distance δ in time t_0.
- Sample p paths to estimate the mean and covariance of X_{t_0}.
- Choose $\bar{b}$, $\bar{\sigma}$ to match the mean and covariance of X_{t_0}.

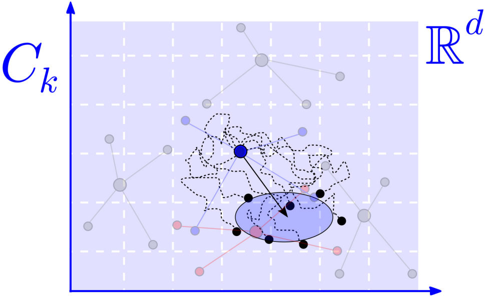


Parameter Estimation

On chart C_k associated with y_k, we fit a constant coefficient SDE

$$d\bar{X}_t = \bar{b}dt + \bar{\sigma}dB_t$$

- Pick t_0 such that X_t moves distance δ in time t_0.
- Sample p paths to estimate the mean and covariance of X_{t_0}.
- Choose $\bar{b}$, $\bar{\sigma}$ to match the mean and covariance of X_{t_0}.

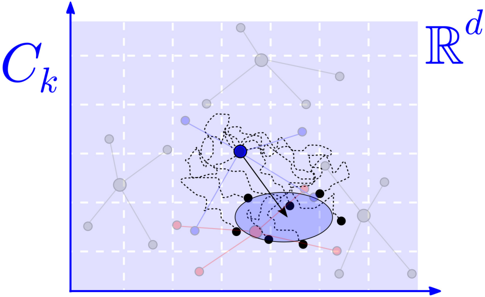


Parameter Estimation

On chart C_k associated with y_k, we fit a constant coefficient SDE

$$d\bar{X}_t = \bar{b}dt + \bar{\sigma}dB_t$$

- Pick t_0 such that X_t moves distance δ in time t_0.
- Sample p paths to estimate the mean and covariance of X_{t_0}.
- Choose $\bar{b}$, $\bar{\sigma}$ to match the mean and covariance of X_{t_0}.



Atlas Simulator

- Forward step

$$x \leftarrow x + \bar{b}_i \Delta t + \bar{\sigma}_i \Delta B$$

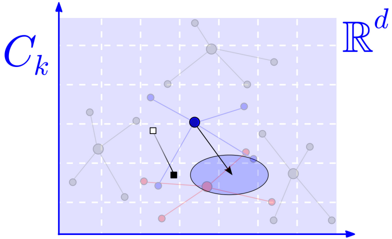
- Chart wall:

$$x \leftarrow W(x)$$

- Transition map

$$i' = \text{argmin}_{j \sim i} |x - \Phi_i(y_j)|$$

$$x \leftarrow S_{i,i'}(x)$$



Atlas Simulator

- Forward step

$$x \leftarrow x + \bar{b}_i \Delta t + \bar{\sigma}_i \Delta B$$

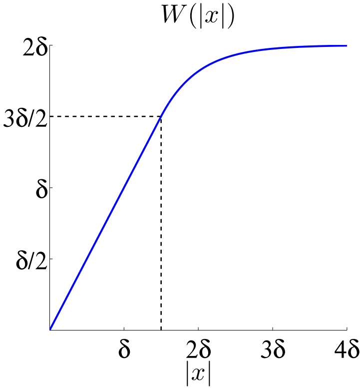
- Chart wall:

$$x \leftarrow W(x)$$

- Transition map

$$i' = \text{argmin}_{j \sim i} |x - \Phi_i(y_j)|$$

$$x \leftarrow S_{i,i'}(x)$$



Atlas Simulator

- Forward step

$$x \leftarrow x + \bar{b}_i \Delta t + \bar{\sigma}_i \Delta B$$

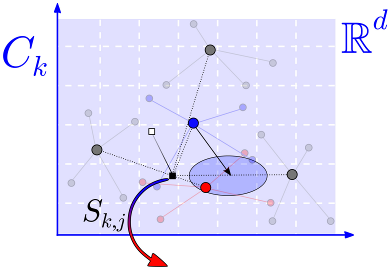
- Chart wall:

$$x \leftarrow W(x)$$

- Transition map

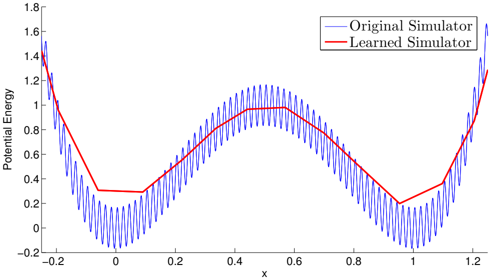
$$i' = \operatorname{argmin}_{j \sim i} |x - \Phi_i(y_j)|$$

$$x \leftarrow S_{i,i'}(x)$$



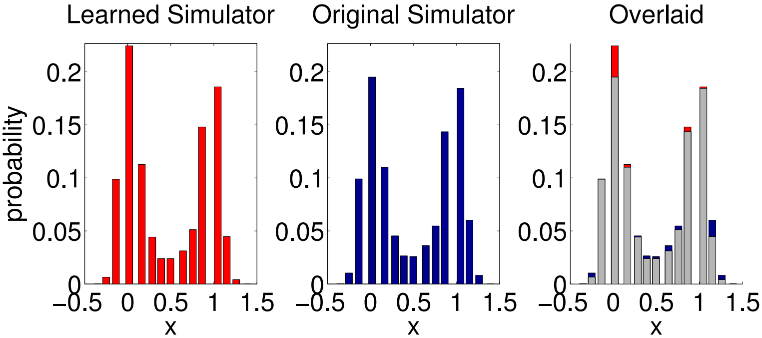
Return to 1-d Example

- $\delta = 0.1$
- $t_0 = 0.01 = 500$ steps
- 14 charts
- $p = 10,000$ samples per chart



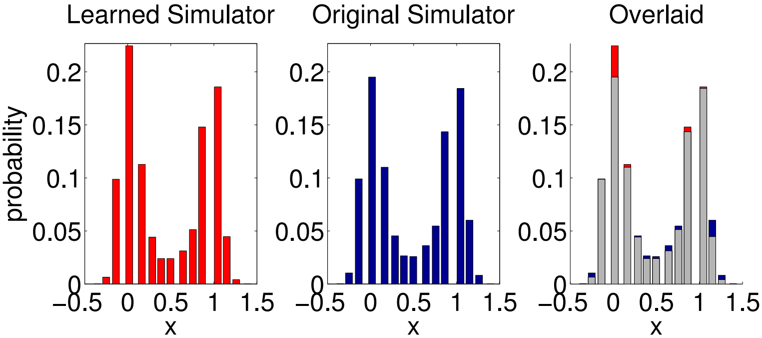
Comparing Simulators

- Pick starting location
- Run many (10,000) trials
- Compare on multiple time scales



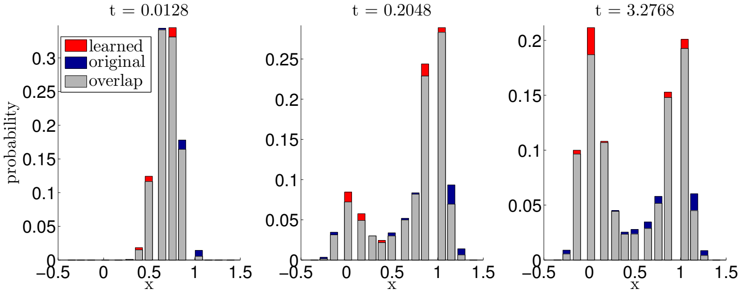
Comparing Simulators

- Pick starting location
- Run many (10,000) trials
- Compare on multiple time scales



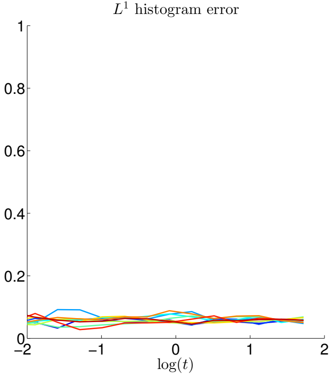
Comparing Simulators

- Pick starting location
- Run many (10,000) trials
- Compare on multiple time scales



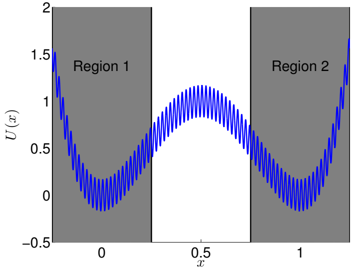
$\mathbb{L}^1$ distance on histograms

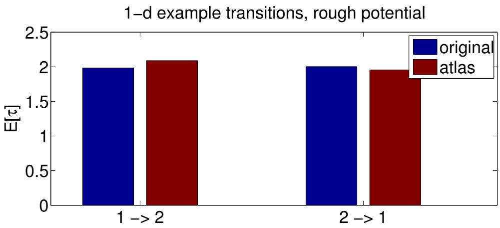
- Pick 10 initial conditions
- Use net for bins
- Run 10,000 paths
- Compare histograms



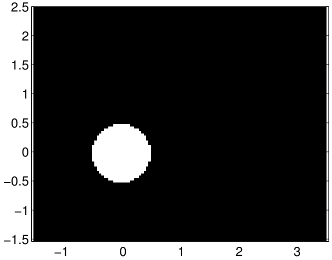
Transition times

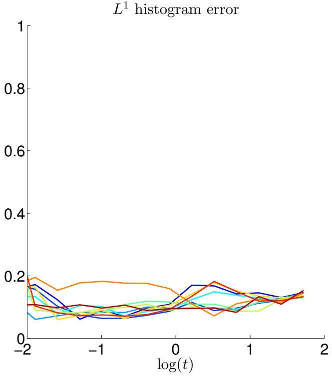
- Define regions (right)
- Measure hitting times





12,500-dim Example: Histogram Comparison

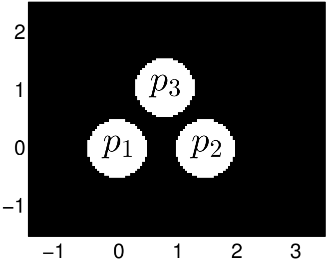


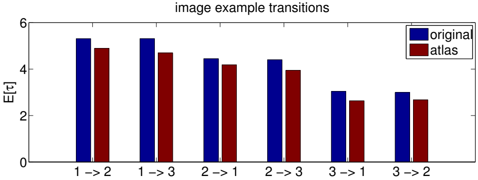


- $t_0 = 0.04 = 800$ steps
- $\delta = 0.2$, ≈ 200 charts
- $2,000$ samples per chart

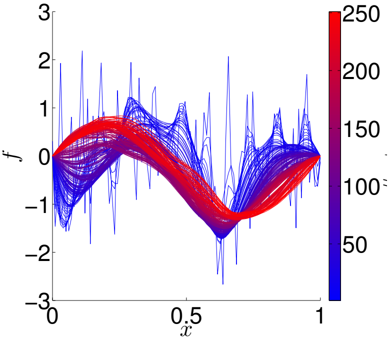
12,500-dim Example: Transition Times

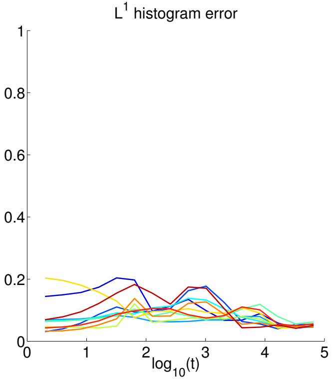
- Region 1: $||p - p_1|| < 1/4$
- Region 2: $||p - p_2|| < 1/4$
- Region 3: $||p - p_3|| < 1/4$





100-dim Example: Histogram Comparison

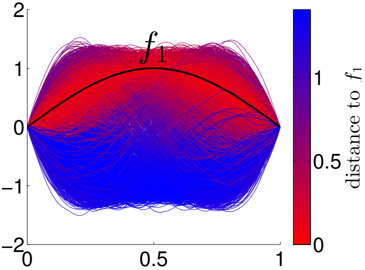


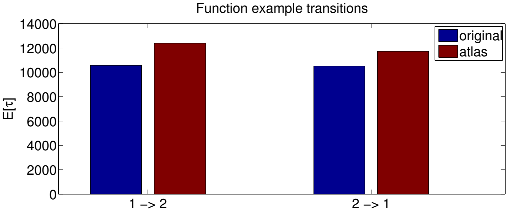


- $t_0 = 0.025 = 250$ steps
- $\delta = 0.3, \approx 70$ charts
- $2,000$ samples per chart

100-dim Example: Transition Times

- Regions: 'up' & 'down'
- Region 1: $||f - f_1|| < 1/2$
- Region 2: $||f + f_1|| < 1/2$





Theorem Statement

Theorem

Let $\mathcal{M}$ be a nice manifold and suppose $X_t \in \mathcal{M}$ solves

$$dX_t = b(X_t)dt + \sigma(X_t)dB_t$$

with b, σ Lipschitz, and σ uniformly elliptic on $\mathcal{M}$. Let q be stationary for X_t and $\widehat{q}$ be stationary for $\widehat{X}_t$. Then if δ is small enough and the number of sample paths $p > (\tau^2 + d)/\delta^4$,

$$||q - \Phi^{-1}(\widehat{q})||_{\mathbb{L}^1(\mathcal{M})} \leq C\delta \log(1/\delta)$$

with probability at least $1 - 2e^{-\tau^2}$.

Future Work

- scale to MD problems
- automatically detecting δ
- visualization software