Bayesian computation in hierarchical models using marginal local approximation MCMC

[1]**Andrew D. Davis**, [1]Youssef Marzouk, [2]Aaron Smith, [3]Natesh Pillai

[1]Department of Aeronautics and Astronautics
Center for Computational Engineering
Massachusetts Institute of Technology
`http://uqgroup.mit.edu`

[2]Department of Mathematics and Statistics
University of Ottawa

[3]Department of Statistics
Harvard University

April 2018

Problem statement

We want to **characterize the distribution π** using a sampling method

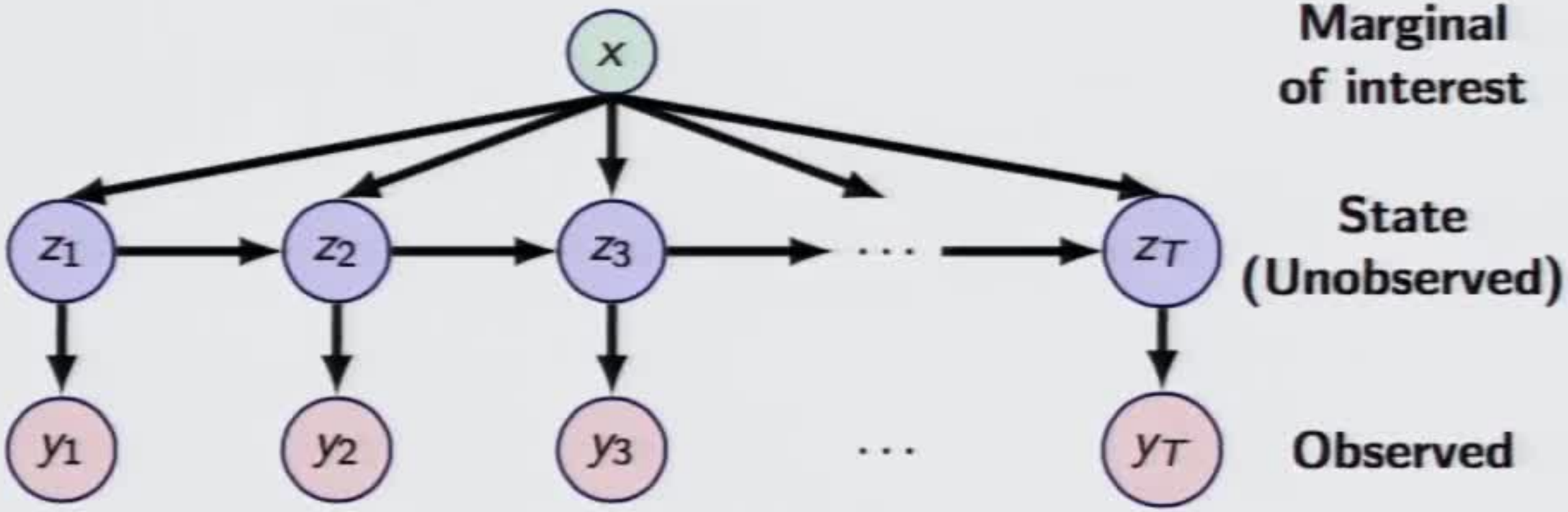
Two problems:

1. The probability density function $\pi(x)$ is **computationally expensive**
2. Only **noisy** density evaluations are available $\tilde{\pi}(x) = \pi(x) + \varepsilon$

Motivating example: state space modelling

Static parameter estimation—

- Given: data $y_{1:T}$ with an unobservable state $z_{1:T}$
- We want to characterize the distribution over $x|y_{1:T}$
  - x is low-dimensional



Local polynomial surrogates

- Given n (potentially noisy) density evaluations at points $\{x^{(i)}\}_{i=1}^{n}$
- Find the degree p polynomial $\hat{\pi}(x)$ that minimizes the weighted least squares error

$$\hat{\pi}(x) = \underset{\rho \in \mathcal{P}_p}{\arg\min} \sum_{i=1}^{n} \left(\rho(x^{(i)}) - \pi(x^{(i)}) \right)^2 K(x^{(i)}, x)$$

  - Locally supported kernel $K(\cdot, x)$

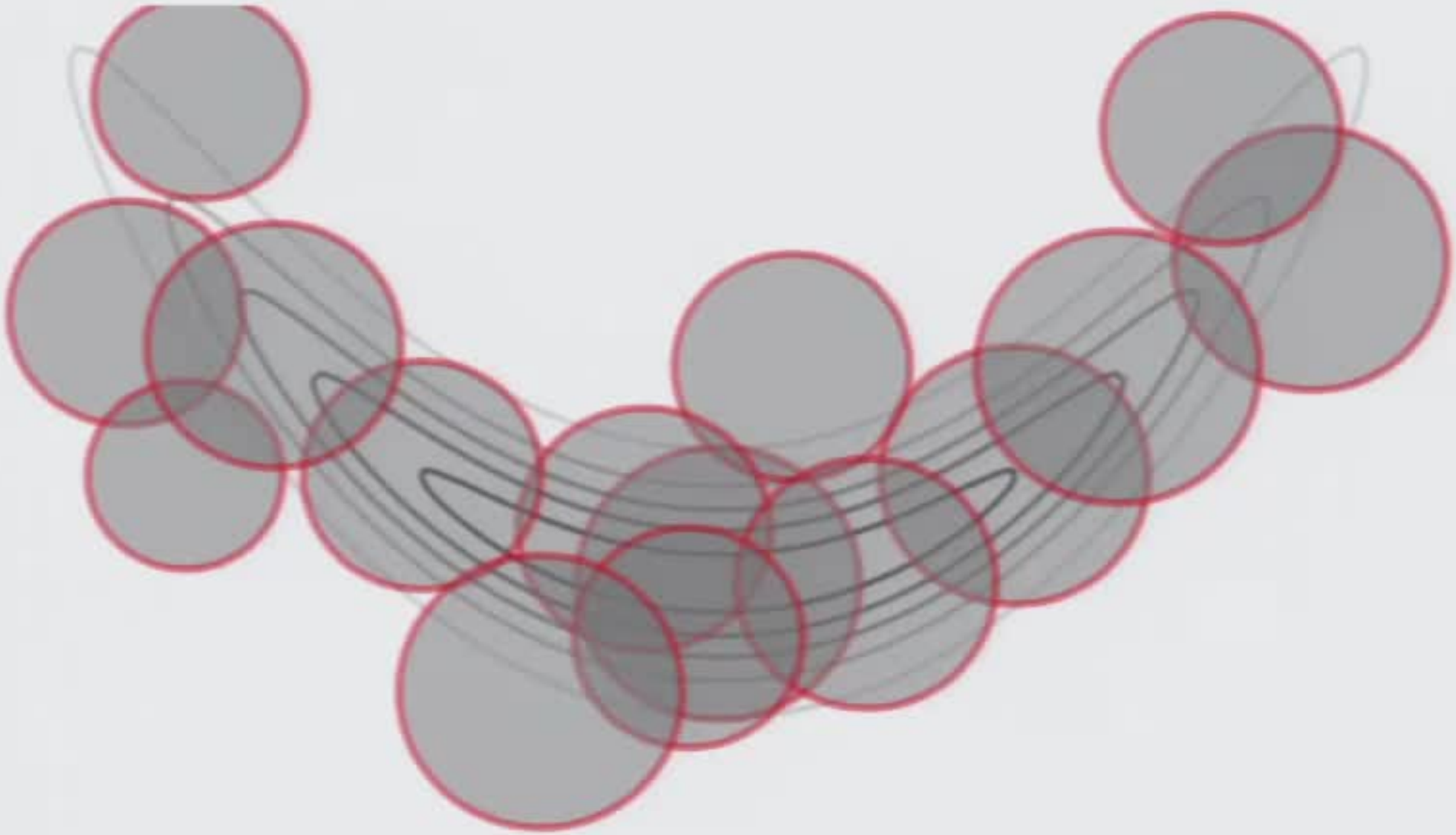
- **Intuition:** minimize the weighted least squares difference between the surrogate and the k nearest neighbors

$\mathcal{B}_k(x)$: smallest ball centered at x with k density evaluations

$$\mathcal{K}(x', x) = \begin{cases} 1 & x' \in \mathcal{B}_k(x) \\ 0 & \text{otherwise} \end{cases}$$

Local polynomial surrogates

Build a local approximation in a ball around each point . . .



Ball size is determined by the prescribed **number of nearest neighbors**

Markov chain Monte Carlo (MCMC) overview

How do we use local polynomial surrogates within MCMC?

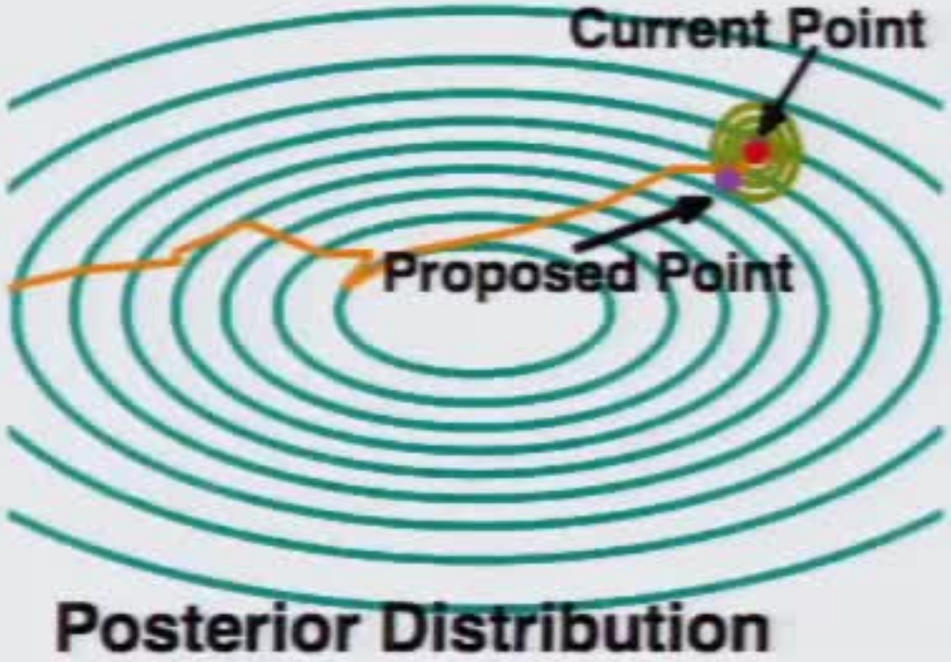
Three step algorithm:

1. Propose $x' \sim p$
2. Acceptance probability

$$\alpha = \min\left(1, \frac{\pi(x')p(x^{(t)}|x')}{\pi(x^{(t)})p(x'|x^{(t)})}\right)$$

3. Accept/reject

$$x^{(t+1)} = \begin{cases} x' & \text{with probability } \alpha \\ x^{(t)} & \text{else} \end{cases}$$



Metropolis et al., 1953

Hastings, 1970

and variations . . .

Haario et al., 2006

Parno and Marzouk, 2014

Brooks et al., 2011

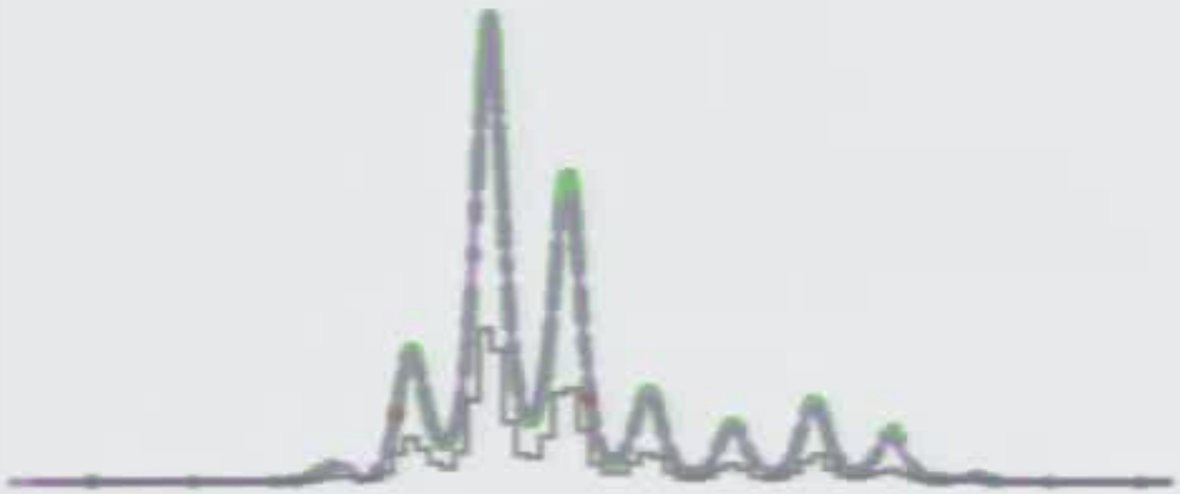
Over versus under refinement

- Choosing the refinement decay rate $0 < \beta_1 < 1$ has a significant effect on the surrogate's quality after a *finite* number of MCMC steps T
  - Note: all $0 < \beta_1 < 1$ are exact when $T \to \infty$

on the surrogate's quality after a *finite* number of MCMC steps T

- Note: all $0 < \beta_1 < 1$ are exact when $T \to \infty$

- Slow decay rate (small β_1) frequently triggers refinement and the estimate's **error is dominated by the MCMC variance**



$\beta_1 = 0.5$

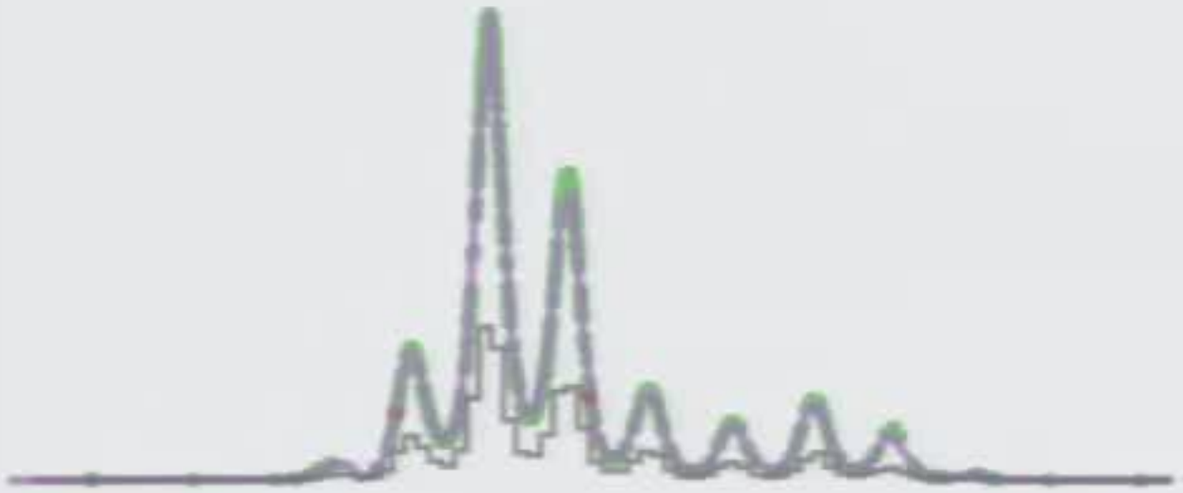
Over versus under refinement

- Choosing the refinement decay rate $0 < \beta_1 < 1$ has a significant effect on the surrogate's quality after a *finite* number of MCMC steps T

on the surrogate's quality after a *finite* number of MCMC steps T

- Note: all $0 < \beta_1 < 1$ are exact when $T \to \infty$

- Slow decay rate (small β_1) frequently triggers refinement and the estimate's **error is dominated by the MCMC variance**



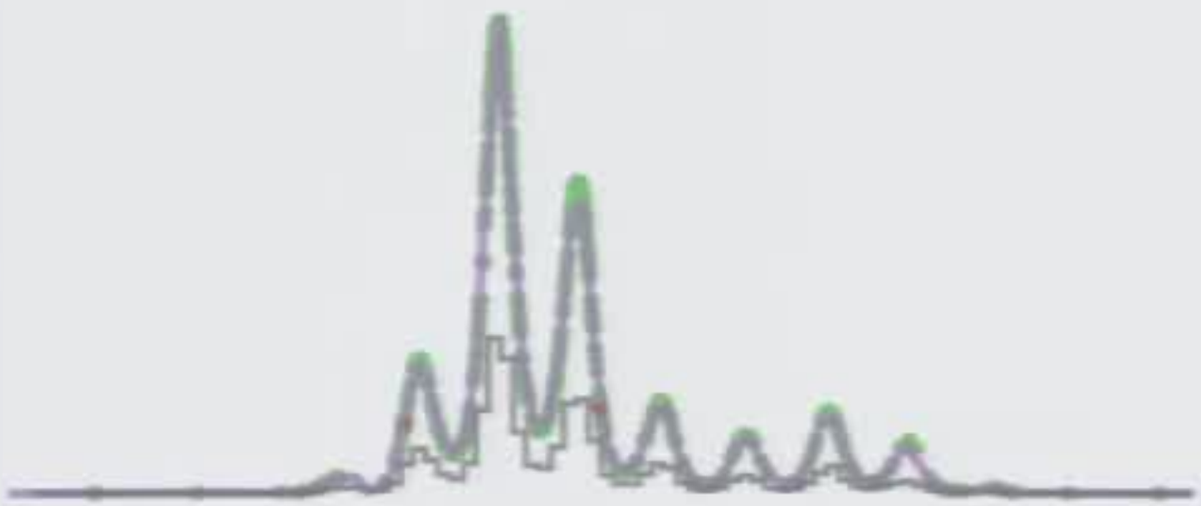
$\beta_1 = 0.5$

Over versus under refinement

- Choosing the refinement decay rate $0 < \beta_1 < 1$ has a significant effect on the surrogate's quality after a *finite* number of MCMC steps T

Over versus under refinement

- Choosing the refinement decay rate $0 < \beta_1 < 1$ has a significant effect on the surrogate's quality after a *finite* number of MCMC steps T
  - Note: all $0 < \beta_1 < 1$ are exact when $T \to \infty$
- Slow decay rate (small β_1) frequently triggers refinement and the estimate's **error is dominated by the MCMC variance**



$\beta_1 = 0.5$

Structural refinement strategy

- We devise a refinement strategy based on a local error estimate to balance MCMC variance and structural error
- Divide the chain into M levels and prescribe an error threshold $\gamma(M) = \gamma_0 M^{-\gamma_1}$ on each level
  - Explore the parameter space before refining the error threshold
- We switch to level $M+1$ at step T_M, when the MCMC variance balances the structural error

MCMC variance	[Structural bias]^2
$C_{MCMC} T_M^{-1}$	$\gamma_0^2 M^{-2\gamma_1}$

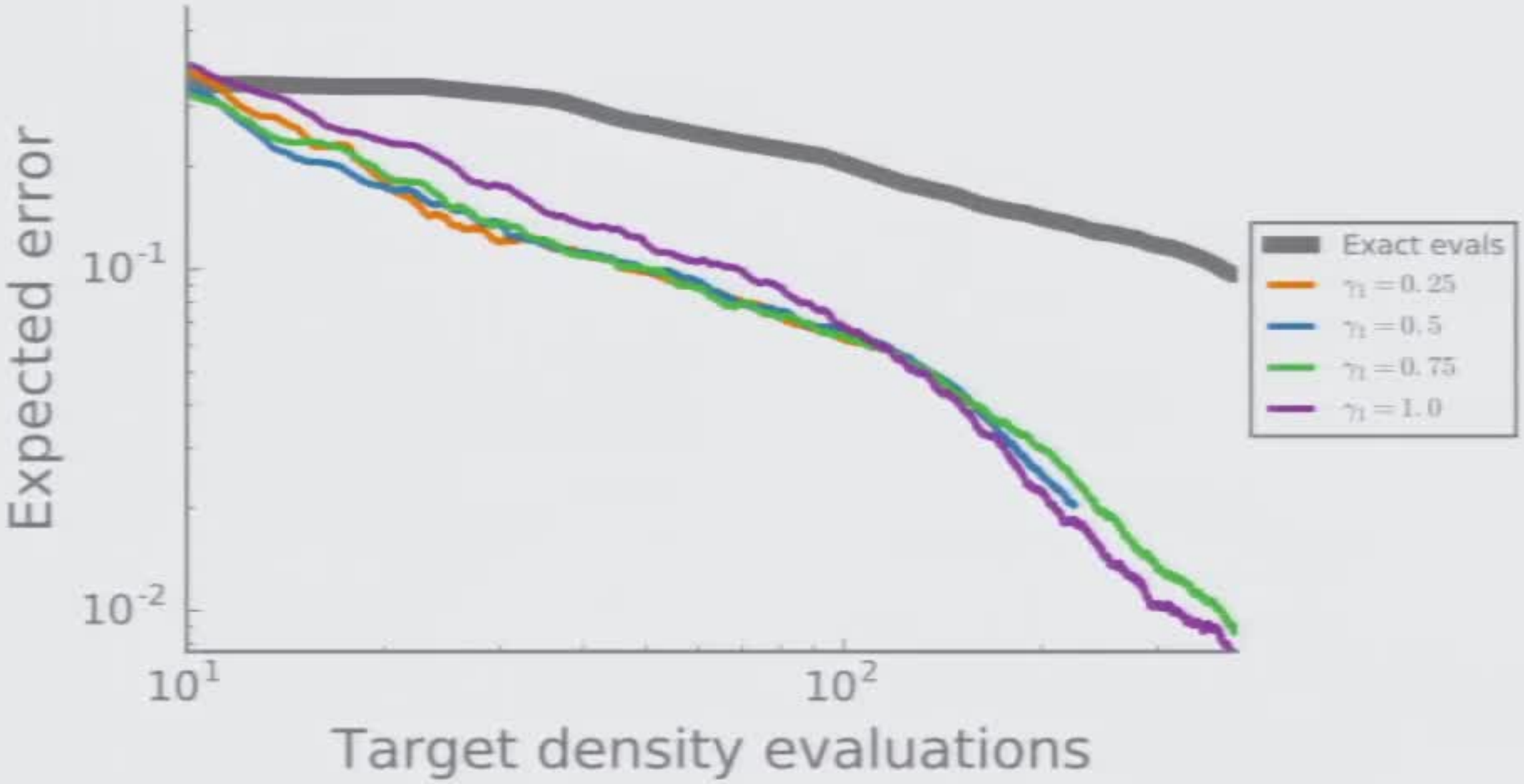
$$T_M = \varphi_1 M^{2\gamma_1}$$

- The threshold decay rate must be $\gamma_1 > 0.5$ so the length of each level $T_M - T_{M-1}$ grows

We trigger refinement based on a piecewise constant error threshold

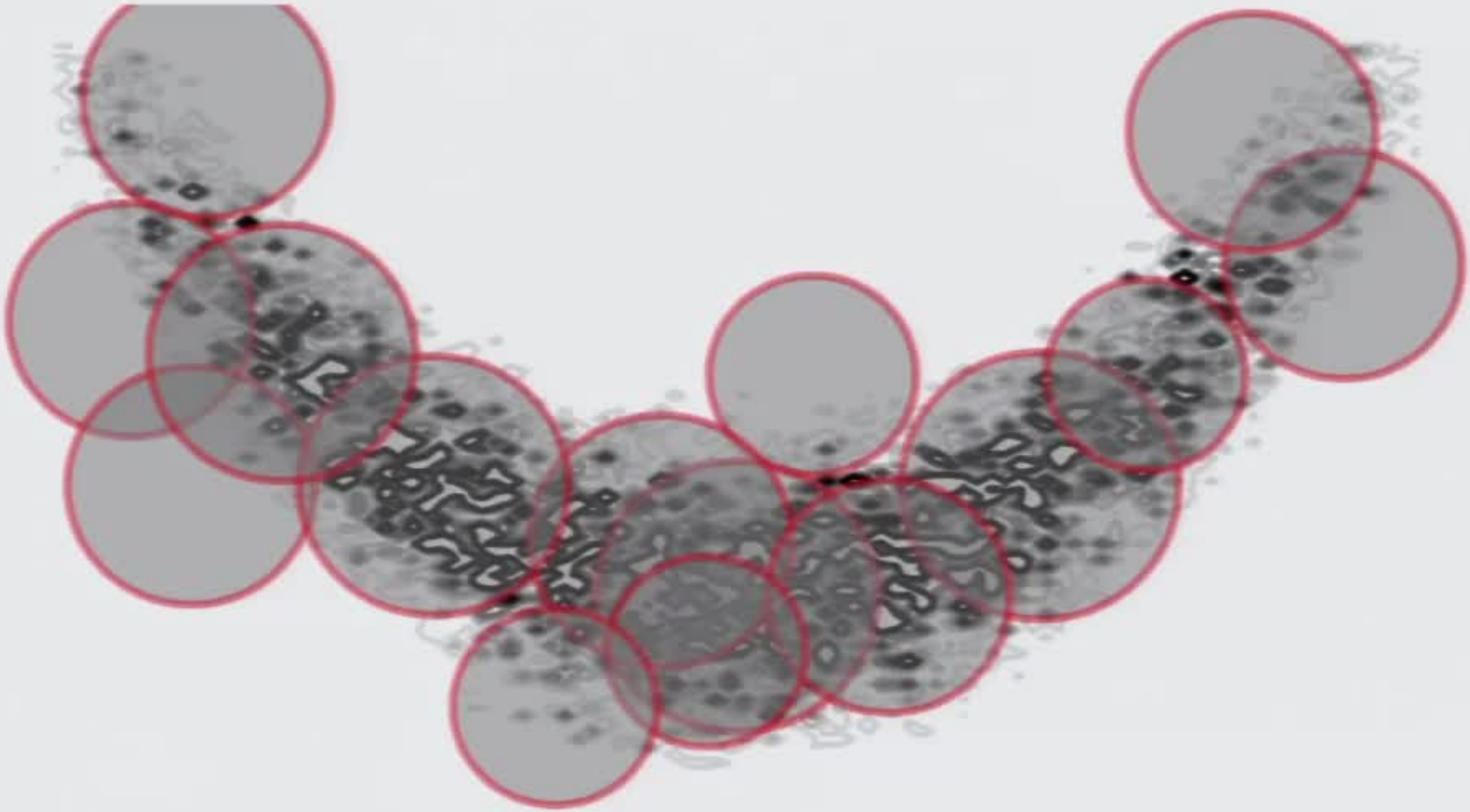
Expected error (structural refinement)

- The expected number of refinements is the same (when $\gamma_1 > 0.5$)
- When $\gamma_1 \leq 0.5$, the surrogate is underrefined
  - The error is dominated by the structural bias



Local polynomial surrogates

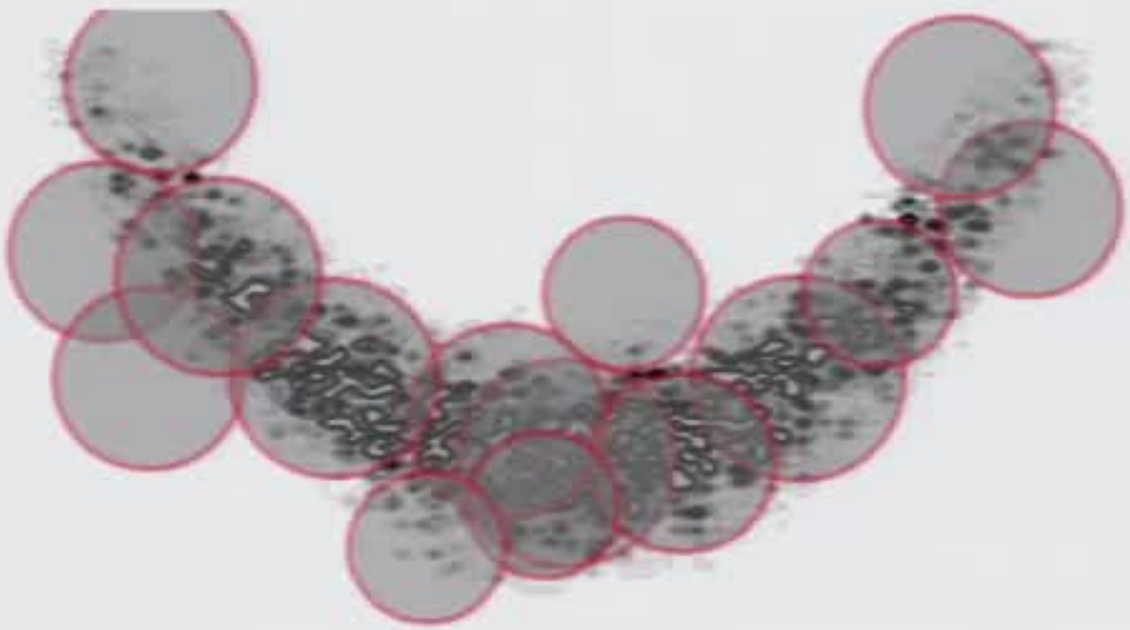
- Now consider situations with **noisy evaluations** of the target density



Surrogate convergence (noisy density evaluations)

In the **noisy evaluation** case, the surrogate $\hat{\pi}(x)$ approaches $\pi(x)$ as:

1. The ball size $\Delta \to 0$ (number of evaluations $n \to \infty$)
2. Λ-poisedness is maintained inside each ball
3. The number of nearest neighbors $k \to \infty$ (while $\frac{k}{n} \to 0$)



Large balls

Small balls

Structural refinement (noisy density evaluations)

At $x^{(t)}$, reset the number of nearest neighbors $k(n) = \lfloor \kappa_0 + \kappa_1 \log(n) \rfloor$ and refine the surrogate if:

1. The poisedness constant is too large $\Lambda^{(t)} > \bar{\Lambda}$
2. If the local error indicator is greater than the level's threshold

$$e(x^{(t)}) = \sqrt{k(n)} \Lambda(x^{(t)}) \Delta^{p+1}(x^{(t)}) > \gamma_0 M^{-\gamma_1}$$

Red dots:

Noisy density evaluations

Purple line:

Local polynomial approximation

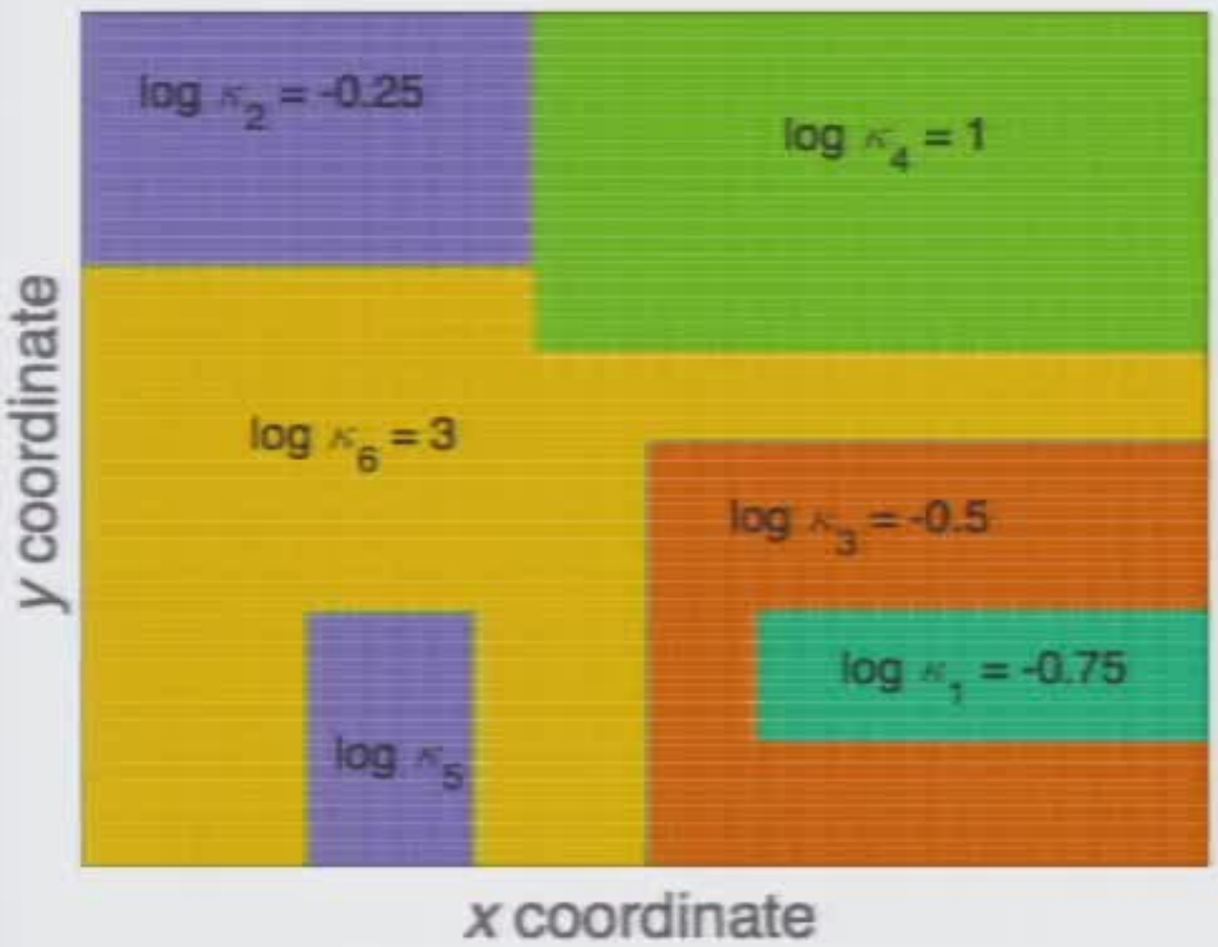
Grey line:

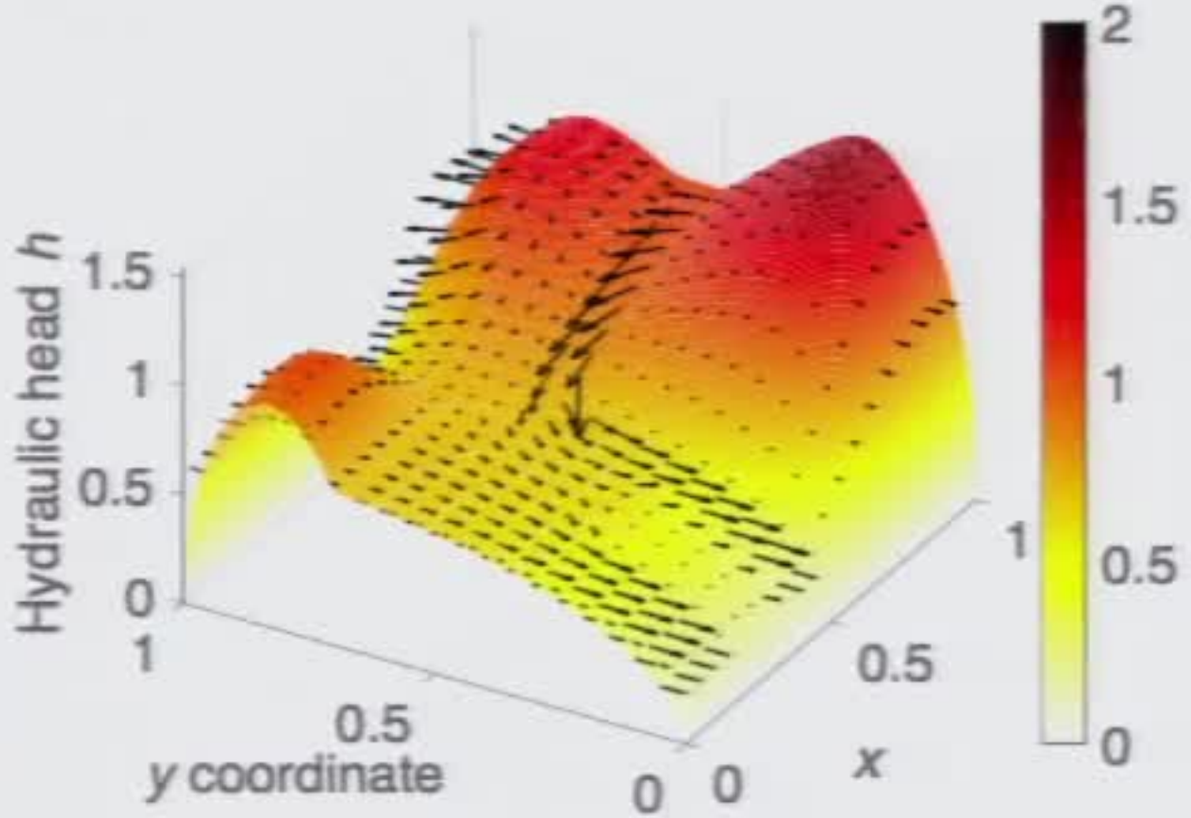
Binned MCMC samples

Tracer transport example: steady state velocities

Compute the steady state hydraulic head and Darcy velocity:

$$\nabla \cdot (\kappa h \nabla h) = f_h \quad \text{and} \quad \begin{bmatrix} u \\ v \end{bmatrix} = -\kappa h \nabla h$$





Our goal is to infer κ_1 κ_2 κ_3 κ_4 κ_5 and κ_6

Conclusions

- Building and refining a local polynomial approximations significantly **reduces computational expense**

Structural refinement (noisy density evaluations)

At $x^{(t)}$, reset the number of nearest neighbors $k(n) = \lfloor \kappa_0 + \kappa_1 \log(n) \rfloor$ and

Large balls

Small balls

Structural refinement (noisy density evaluations)

At $x^{(t)}$, reset the number of nearest neighbors $k(n) = \lfloor \kappa_0 + \kappa_1 \log(n) \rfloor$ and

Structural refinement (noisy density evaluations)

At $x^{(t)}$, reset the number of nearest neighbors $k(n) = \lfloor \kappa_0 + \kappa_1 \log(n) \rfloor$ and refine the surrogate if:

1. The poisedness constant is too large $\Lambda^{(t)} > \bar{\Lambda}$
2. If the local error indicator is greater than the level's threshold

$$e(x^{(t)}) = \sqrt{k(n)}\Lambda(x^{(t)})\Delta^{p+1}(x^{(t)}) > \gamma_0 M^{-\gamma_1}$$

Structural refinement (noisy density evaluations)

At $x^{(t)}$, reset the number of nearest neighbors $k(n) = \lfloor \kappa_0 + \kappa_1 \log(n) \rfloor$ and refine the surrogate if:

1. The poisedness constant is too large $\Lambda^{(t)} > \bar{\Lambda}$
2. If the local error indicator is greater than the level's threshold

$$e(x^{(t)}) = \sqrt{k(n)}\Lambda(x^{(t)})\Delta^{p+1}(x^{(t)}) > \gamma_0 M^{-\gamma_1}$$

Structural refinement (noisy density evaluations)

At $x^{(t)}$, reset the number of nearest neighbors $k(n) = \lfloor \kappa_0 + \kappa_1 \log(n) \rfloor$ and refine the surrogate if:

1. The poisedness constant is too large $\Lambda^{(t)} > \bar{\Lambda}$
2. If the local error indicator is greater than the level's threshold

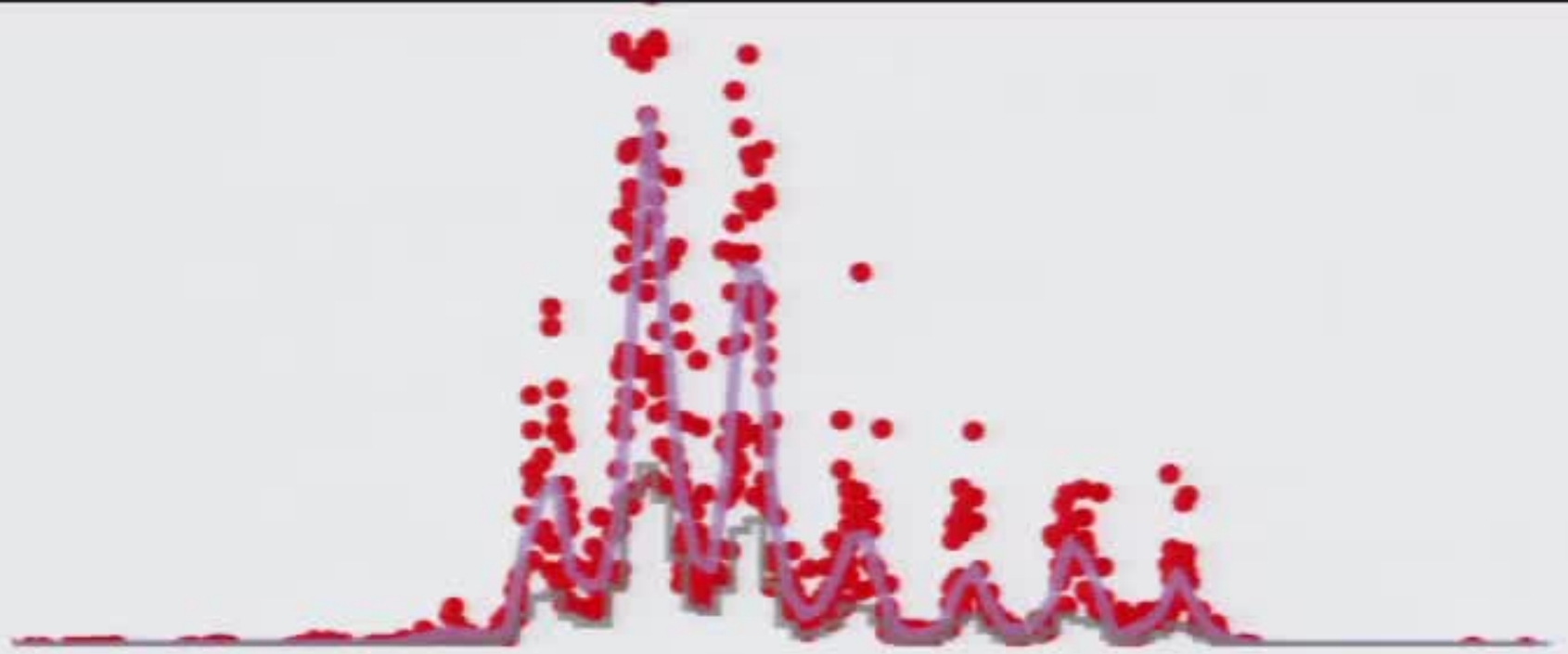
$$e(x^{(t)}) = \sqrt{k(n)}\Lambda(x^{(t)})\Delta^{p+1}(x^{(t)}) > \gamma_0 M^{-\gamma_1}$$

Structural refinement (noisy density evaluations)

At $x^{(t)}$, reset the number of nearest neighbors $k(n) = \lfloor \kappa_0 + \kappa_1 \log(n) \rfloor$ and refine the surrogate if:

1. The poisedness constant is too large $\Lambda^{(t)} > \bar{\Lambda}$
2. If the local error indicator is greater than the level's threshold

$$e(x^{(t)}) = \sqrt{k(n)}\Lambda(x^{(t)})\Delta^{p+1}(x^{(t)}) > \gamma_0 M^{-\gamma_1}$$



evaluations

Purple line:

Local polynomial approximation

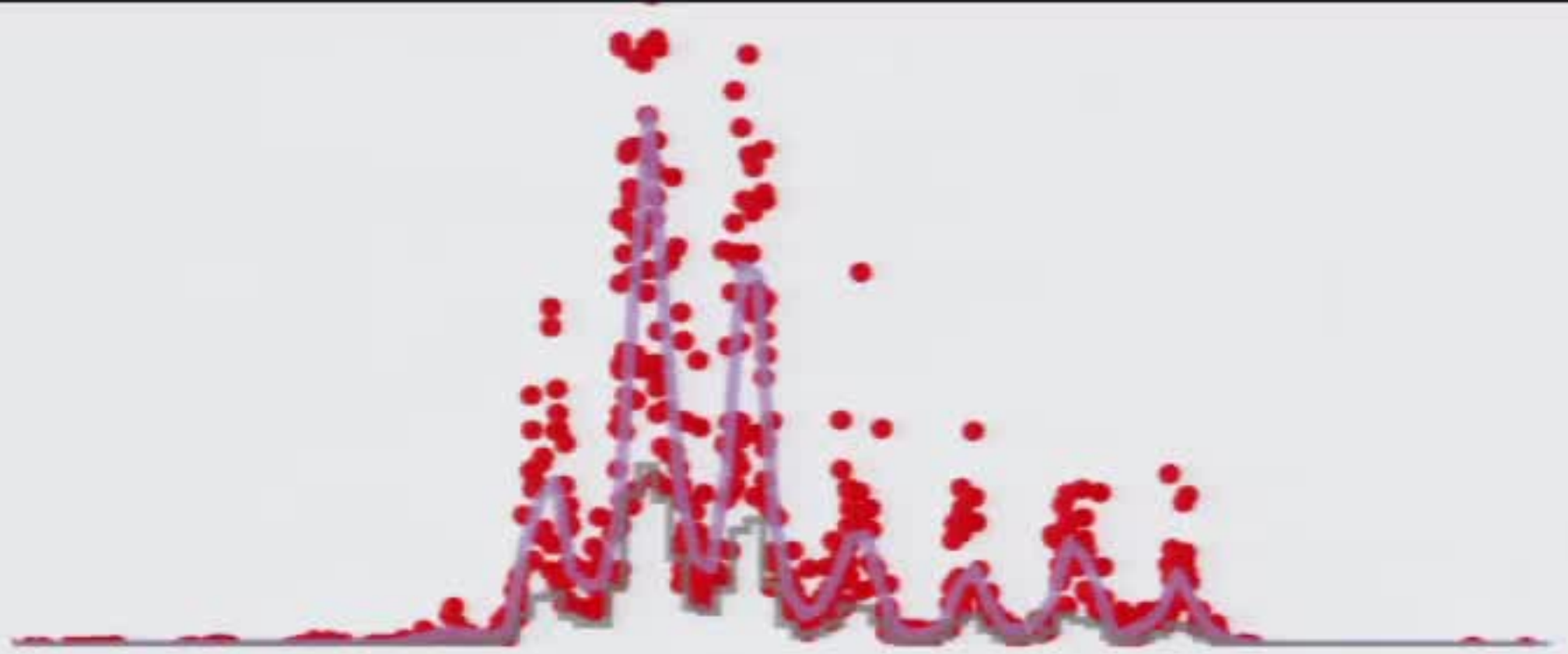
Grey line:

Binned MCMC samples

Tracer transport example: steady state velocities

Compute the steady state hydraulic head and Darcy velocity:

$$\nabla \cdot (\kappa h \nabla h) = f_h \quad \text{and} \quad \begin{bmatrix} u \\ v \end{bmatrix} = -\kappa h \nabla h$$



evaluations

Purple line:

Local polynomial approximation

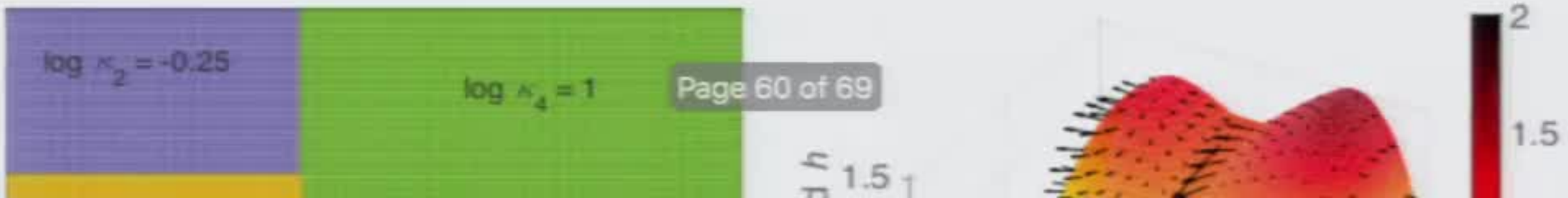
Grey line:

Binned MCMC samples

Tracer transport example: steady state velocities

Compute the steady state hydraulic head and Darcy velocity:

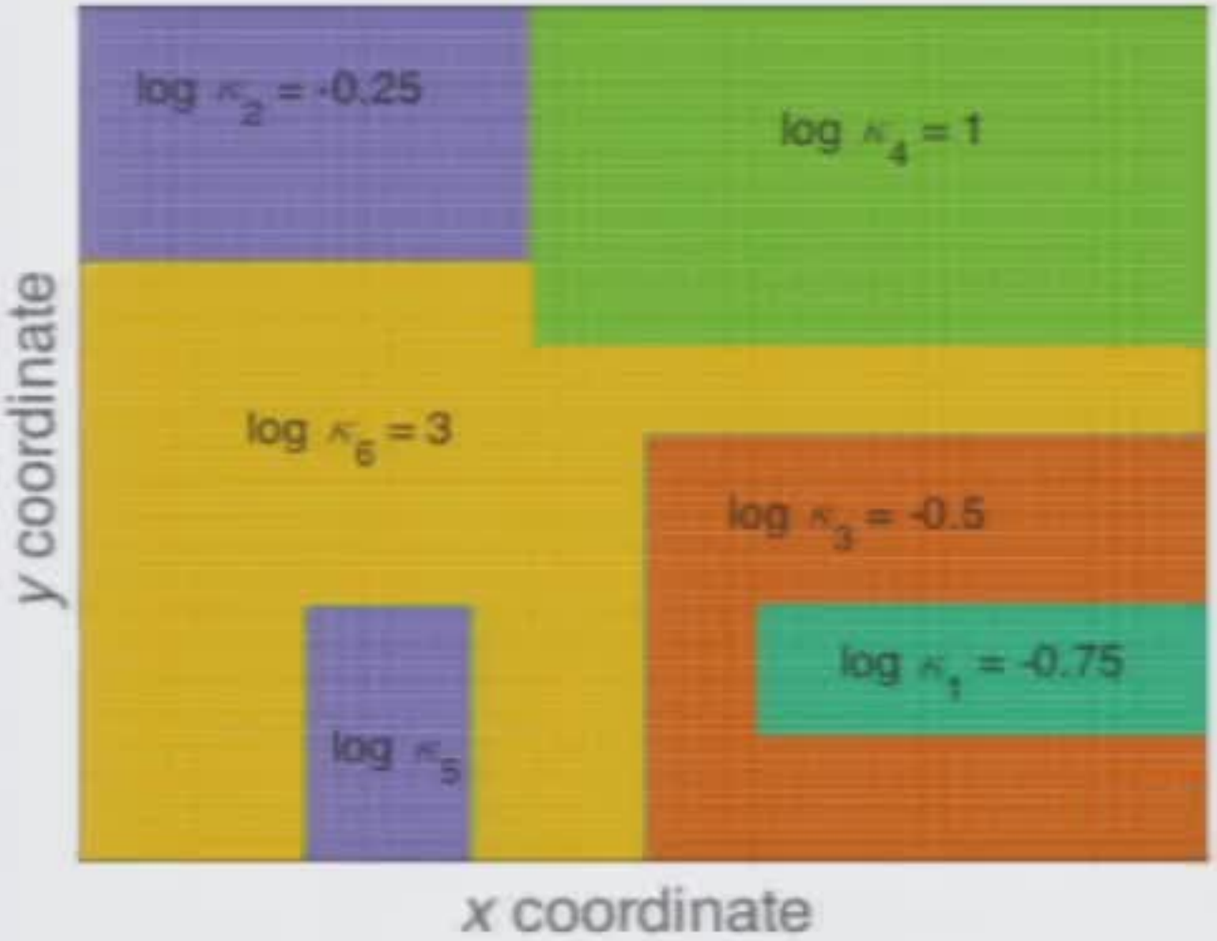
$$\nabla \cdot (\kappa h \nabla h) = f_h \quad \text{and} \quad \begin{bmatrix} u \\ v \end{bmatrix} = -\kappa h \nabla h$$

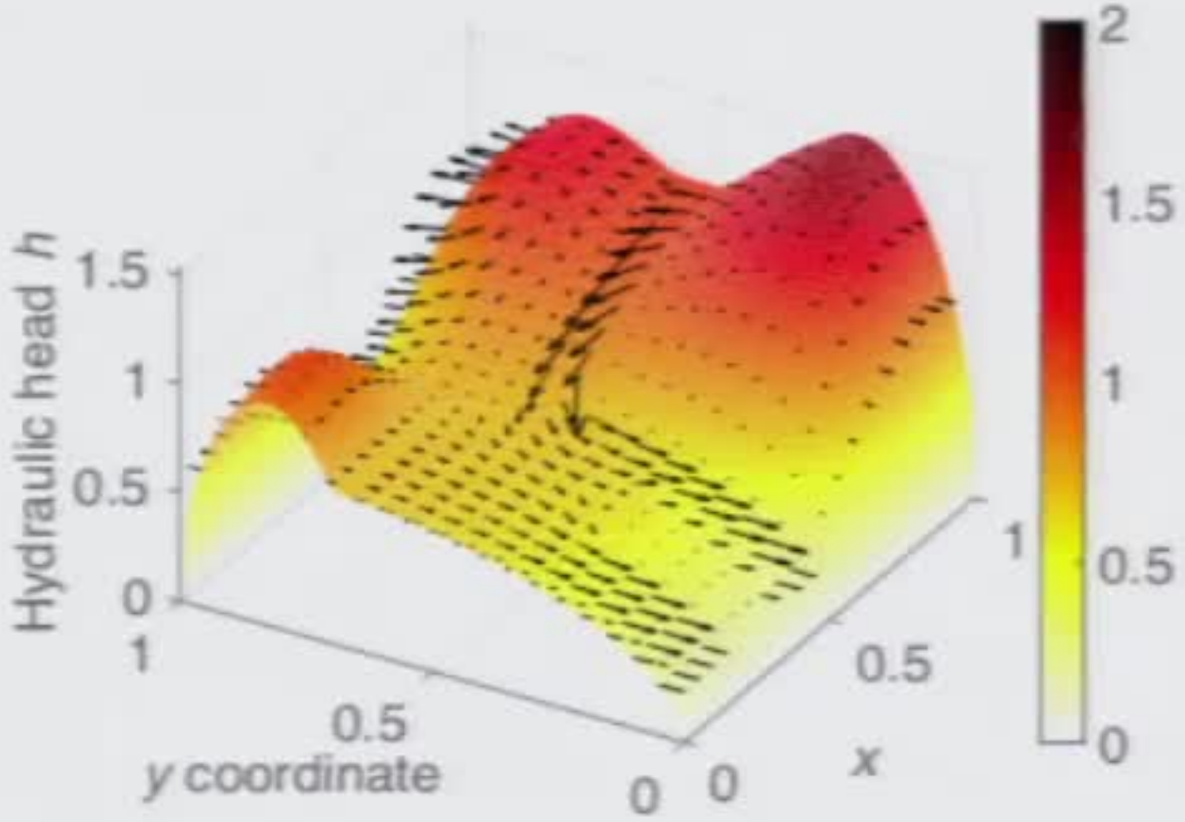


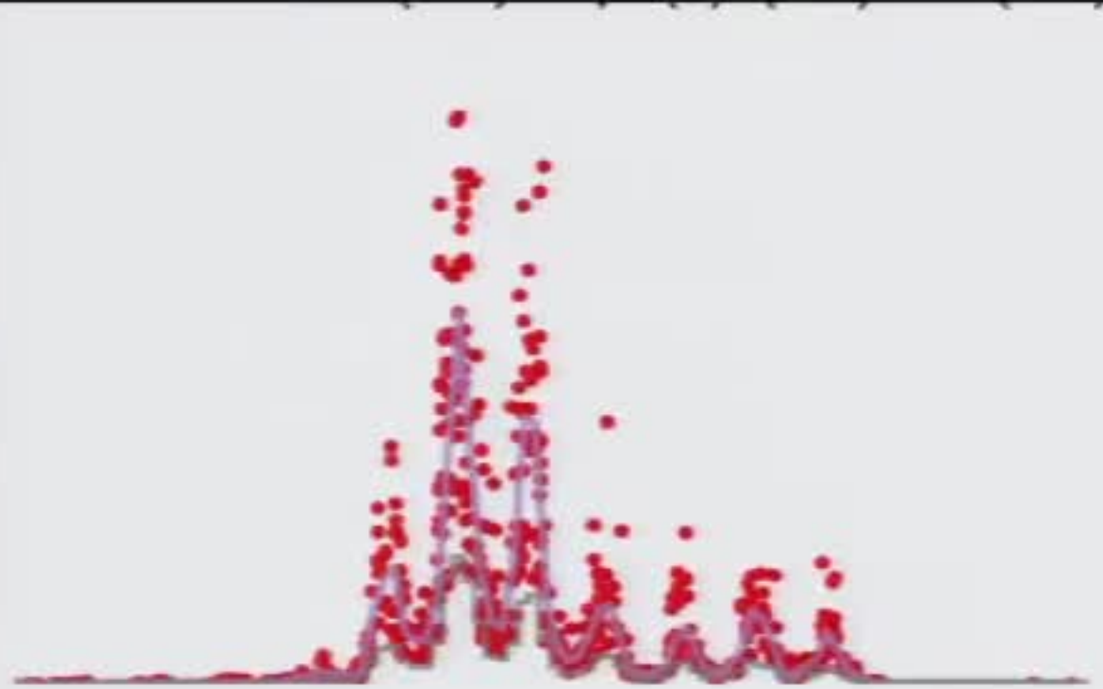
Tracer transport example: steady state velocities

Compute the steady state hydraulic head and Darcy velocity:

$$\nabla \cdot (\kappa h \nabla h) = f_h \quad \text{and} \quad \begin{bmatrix} u \\ v \end{bmatrix} = -\kappa h \nabla h$$







Red dots:

Noisy density evaluations

Purple line:

Local polynomial approximation

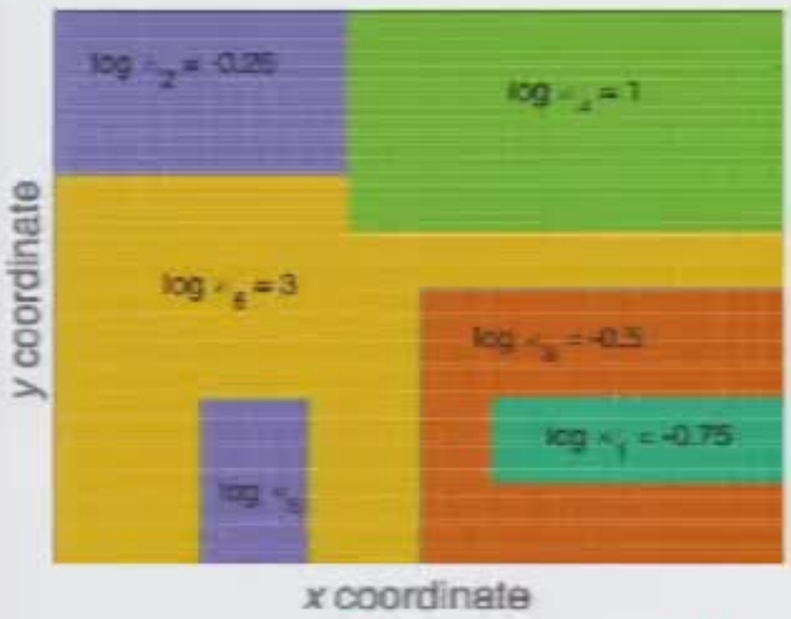
Grey line:

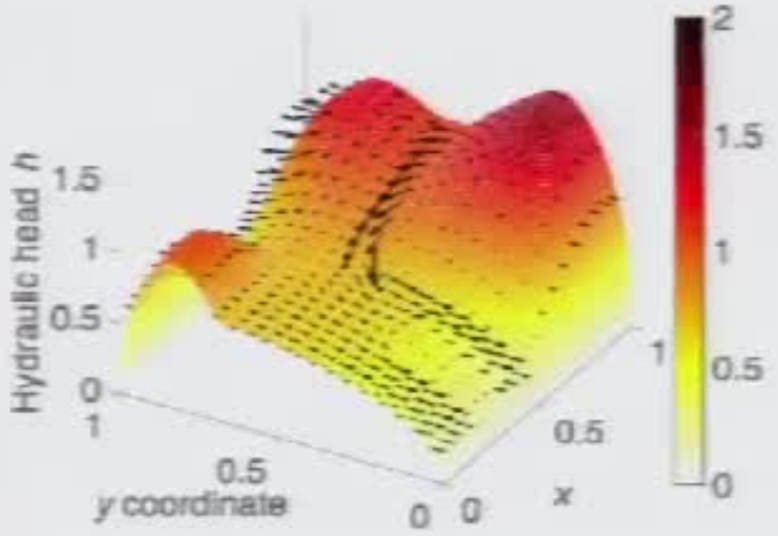
Binned MCMC samples

Tracer transport example: steady state velocities

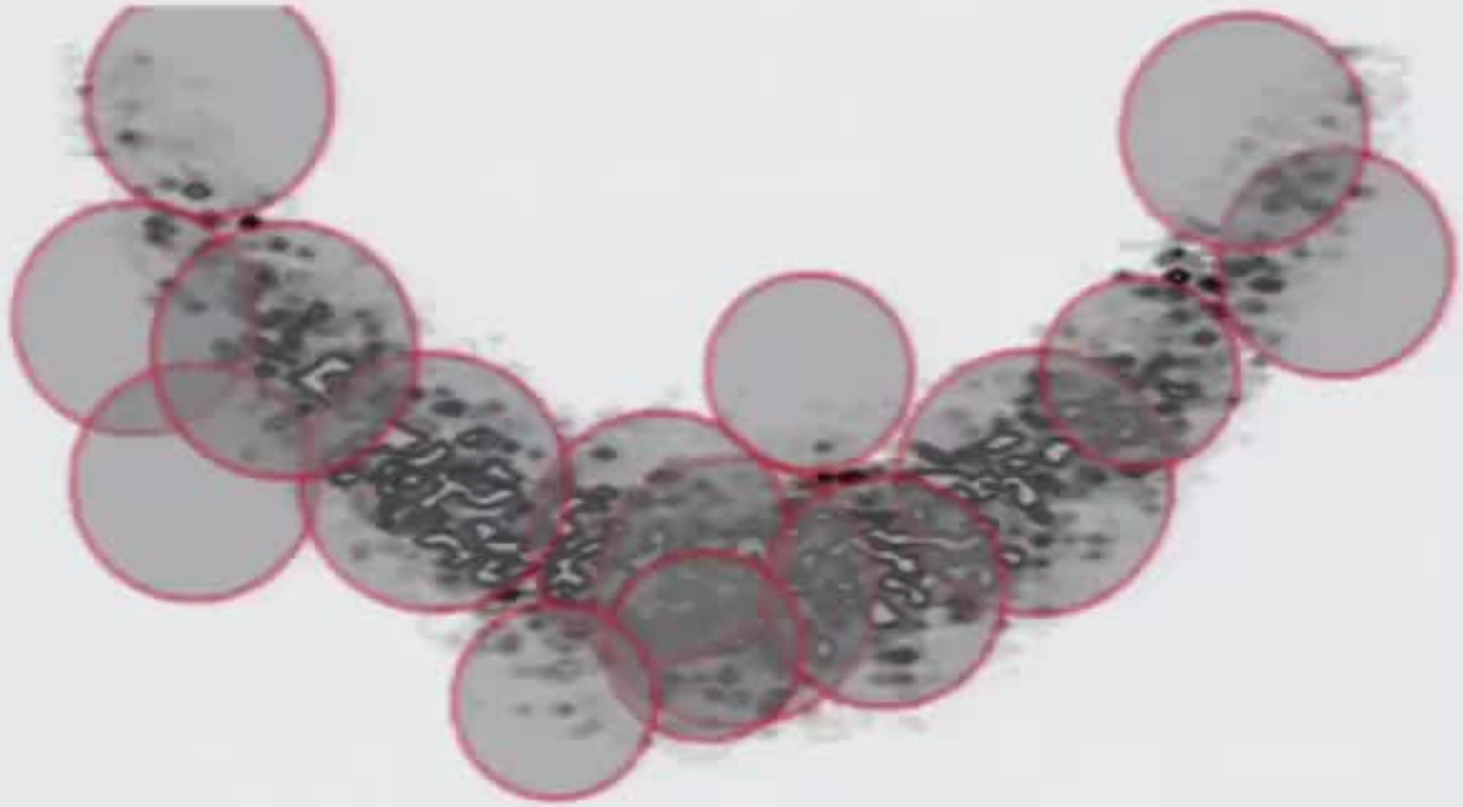
Compute the steady state hydraulic head and Darcy velocity:

$$\nabla \cdot (\kappa h \nabla h) = f_h \quad \text{and} \quad \begin{bmatrix} u \\ v \end{bmatrix} = -\kappa h \nabla h$$





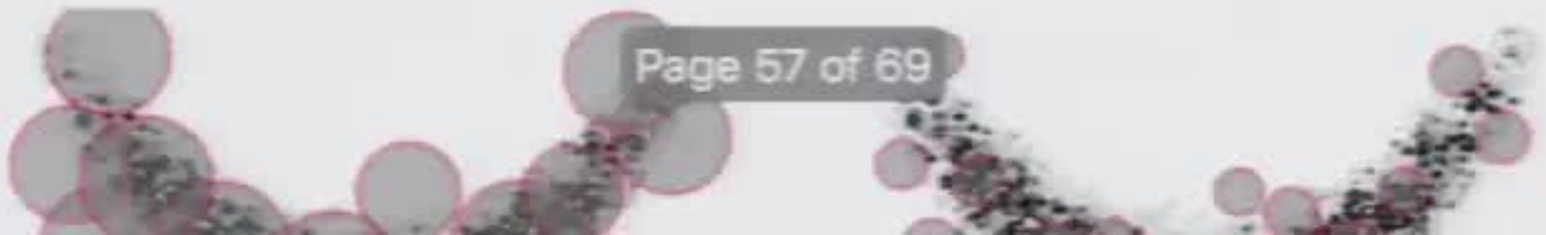
- Now consider situations with **noisy evaluations** of the target density



Surrogate convergence (noisy density evaluations)

In the **noisy evaluation** case, the surrogate $\hat{\pi}(x)$ approaches $\pi(x)$ as:

1. The ball size $\Delta \to 0$ (number of evaluations $n \to \infty$)
2. Λ-poisedness is maintained inside each ball
3. The number of nearest neighbors $k \to \infty$ (while $\frac{k}{n} \to 0$)



An ideal refinement rate

- Assume the MCMC variance decays with the number of steps

$$[\text{MCMC variance}] \leq C_{MCMC} T^{-1}$$

- The surrogate bias is bounded

$$[\text{Surrogate bias}] = |\hat{\pi}(x) - \pi(x)| \leq C_{Surrogate} \sqrt{k} \bar{\Lambda} \Delta^{p+1}$$

- Ideally, we balance MCMC variance with surrogate bias squared to derive an ideal refinement rate

$$C_{MCMC} T^{-1} \sim C_{Surrogate}^2 k \bar{\Lambda}^2 \Delta^{2(p+1)}$$

Structural refinement strategy

- We devise a refinement strategy based on a local error estimate to balance MCMC variance and structural error

An ideal refinement rate

- Assume the MCMC variance decays with the number of steps

$$[\text{MCMC variance}] \leq C_{MCMC} T^{-1}$$

Quantifying poisedness

- Randomly chosen points form clusters
  - Subsets of k nearest neighbors are poorly poised
  - We swap an existing point to improve poisedness

Quantifying poisedness

squares error

$$\hat{\pi}(x) = \arg\min_{\rho \in \mathcal{P}_p} \sum_{i=1}^{n} \left(\rho(x^{(i)}) - \pi(x^{(i)}) \right)^2 K(x^{(i)}, x)$$

- Locally supported kernel $K(\cdot, x)$

Local polynomial surrogates

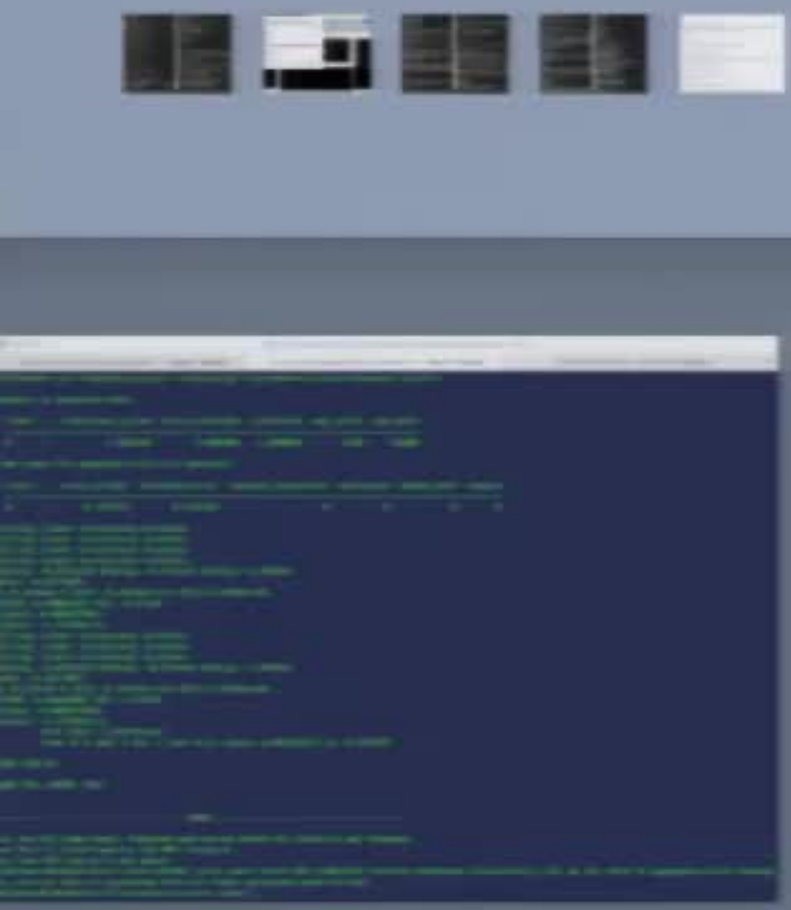
- Given n (potentially noisy) density evaluations at points $\{x^{(i)}\}_{i=1}^n$
- Find the degree p polynomial $\hat{\pi}(x)$ that minimizes the weighted least squares error

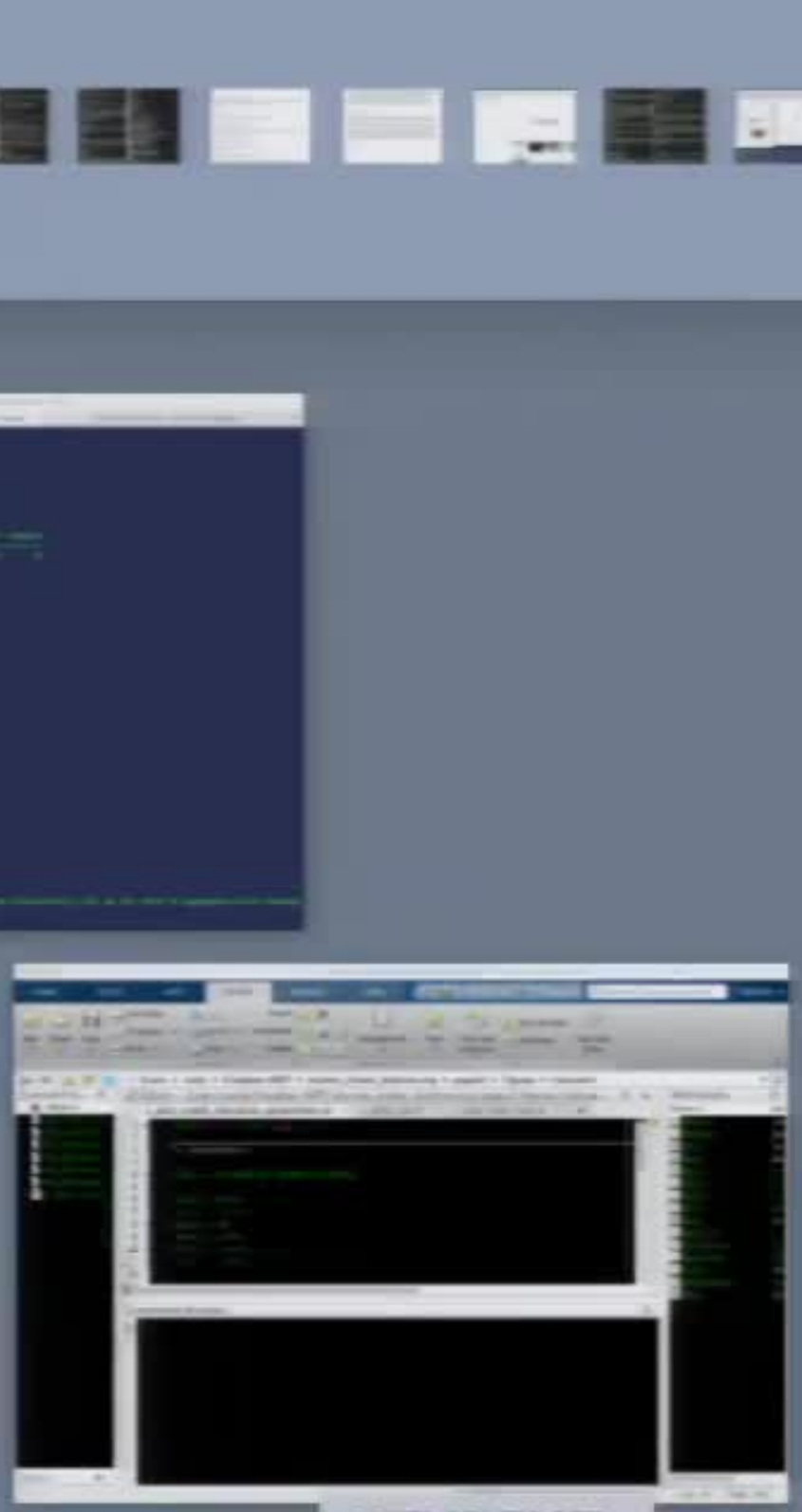
$$\hat{\pi}(x) = \arg\min_{\rho \in \mathcal{P}_p} \sum_{i=1}^{n} \left(\rho(x^{(i)}) - \pi(x^{(i)}) \right)^2 K(x^{(i)}, x)$$

  - Locally supported kernel $K(\cdot, x)$

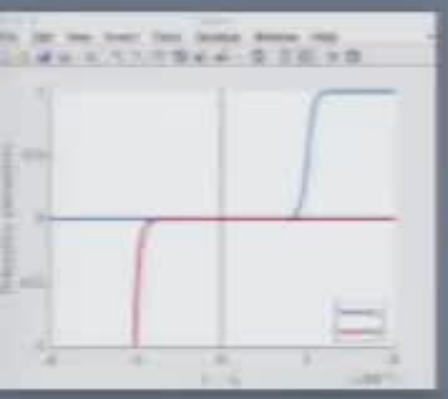
- **Intuition:** minimize the weighted least squares difference between the surrogate and the k near

$\mathcal{B}_k(x)$: smallest ball centered at x with k density evaluations





MATLAB_R2017a



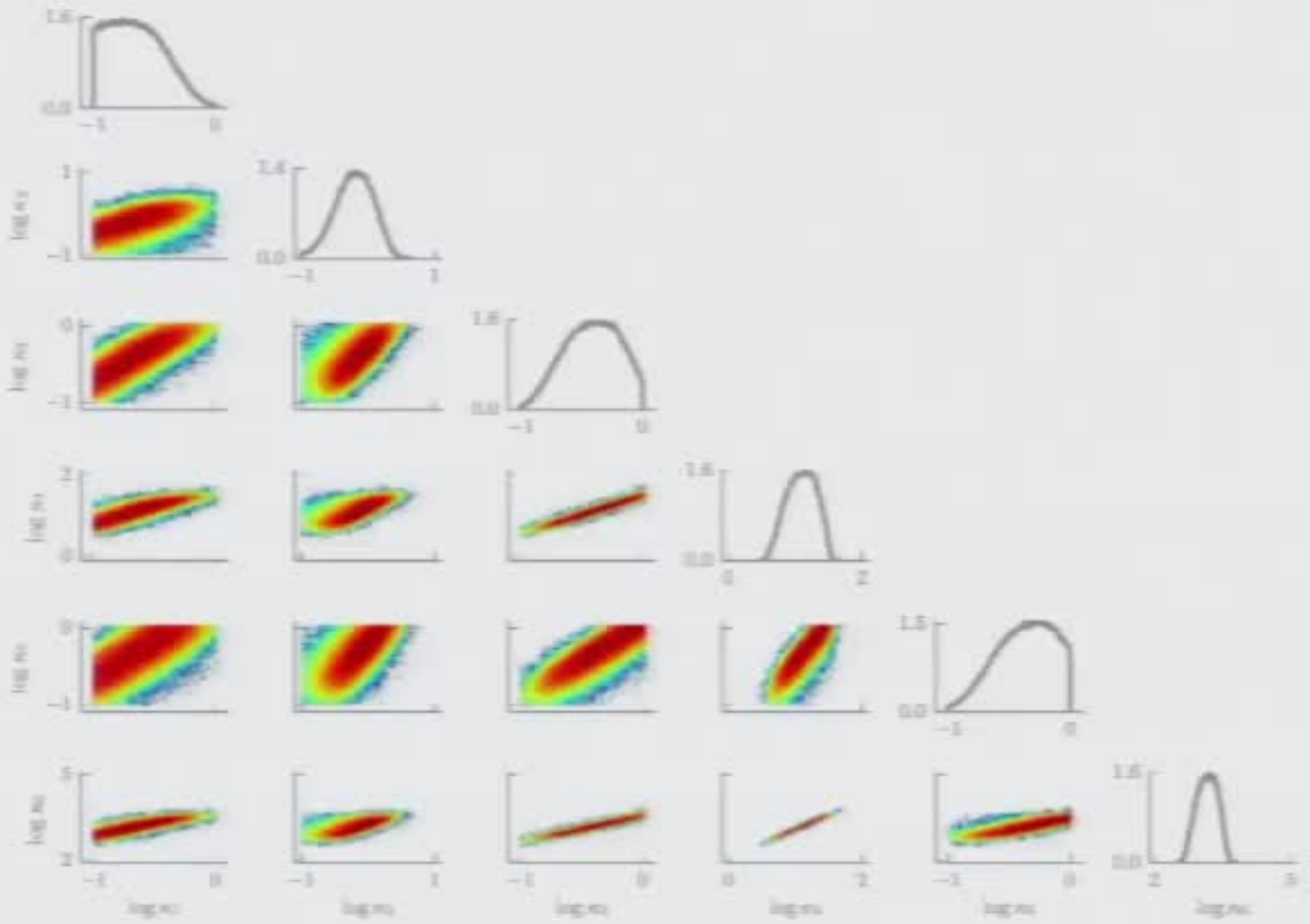


Figure 9. *One- and two-dimensional posterior marginals of the parameters in the hydrologic tracer transport problem. Bounds on each subplot axis are the upper and lower bounds for the uniform prior on the corresponding parameter (Table 1).*

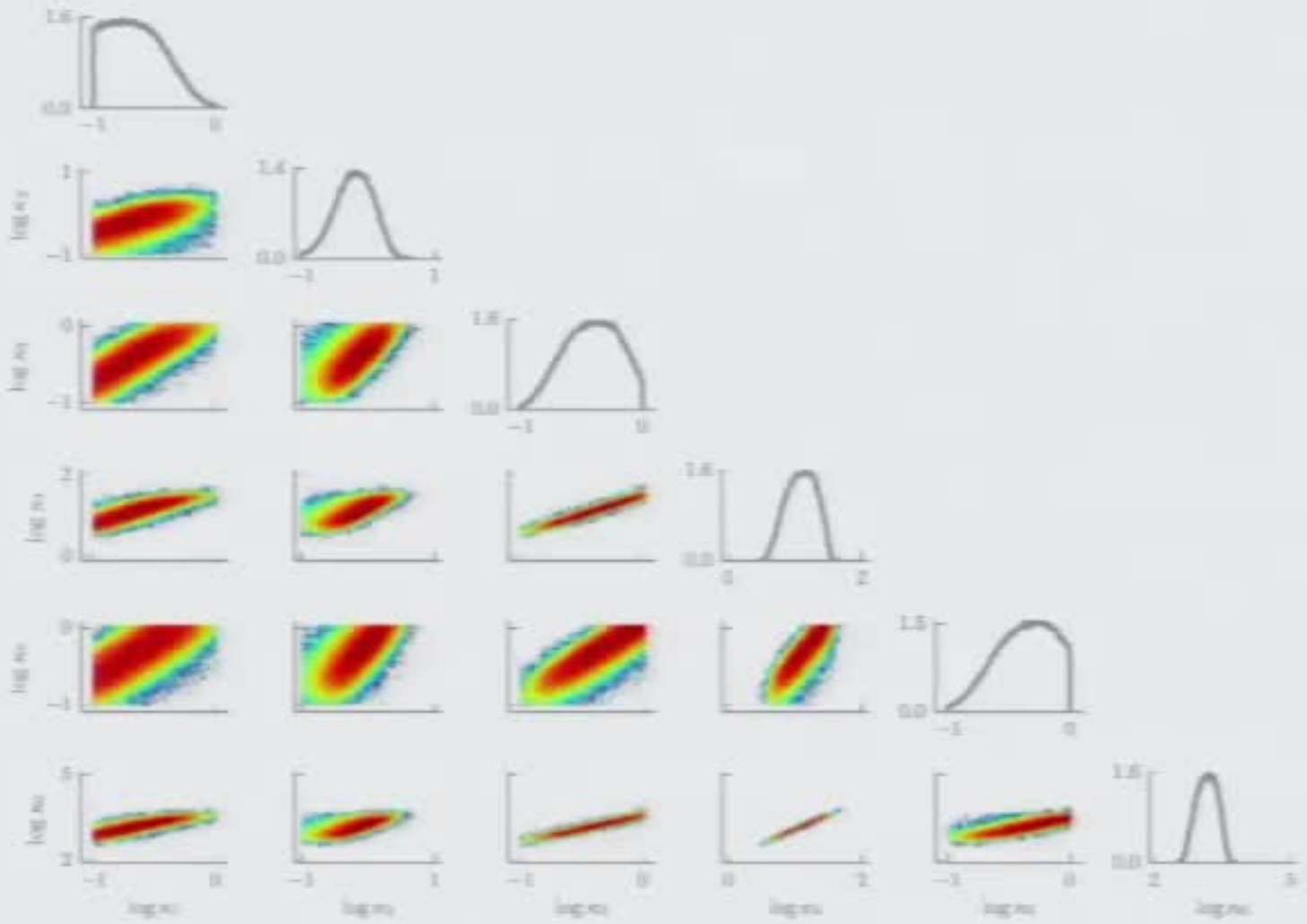


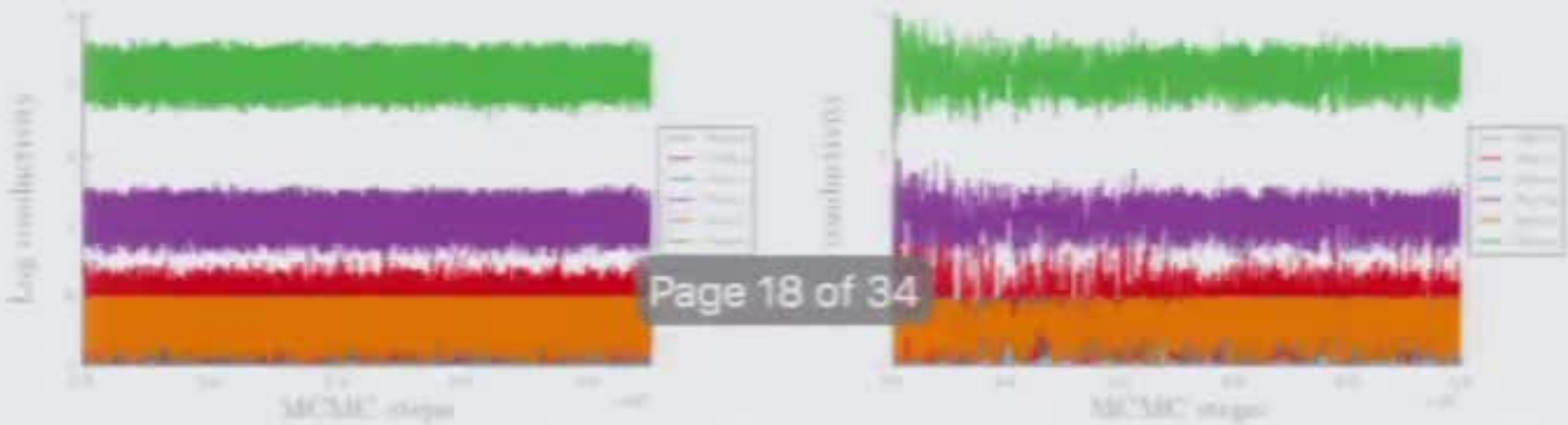
Figure 9. *One- and two-dimensional posterior marginals of the parameters in the hydrologic tracer transport problem. Bounds on each subplot axis are the upper and lower bounds for the uniform prior on the corresponding parameter (Table 1).*

y coordinate

Figure 7. *The tracer concentration $c(x, y, t = 0.4)$, given the conductivity field in Figure 5. The tracer is injected from a well in each corner.*

SIAM/ASA J. UNCERTAINTY QUANTIFICATION
Vol. xx, pp. x

local approximations, we also wish to compare our approach with chains that employ exact evaluations of the forward model. To make such comparisons feasible—and also to reflect computational practice for complex PDE models—we parallelize each forward model evaluation. We use four processors, which reduces the forward model's runtime to roughly 4 seconds of computation. Thus our parallel MCMC scheme actually employs two levels of parallelism: an outer level involving parallel chains, as described in Section 3, and an inner level within each forward model evaluation.



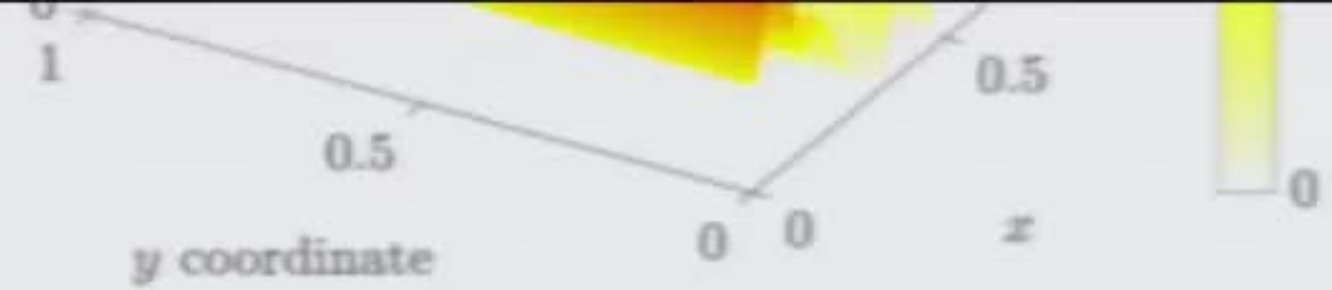


Figure 7. *The tracer concentration $c(x, y, t = 0.4)$, given the conductivity field in Figure 5. The tracer is injected from a well in each corner.*

local approximations, we also wish to compare our approach with chains that employ exact evaluations of the forward model. To make such comparisons feasible—and also to reflect computational practice for complex PDE models—we parallelize each forward model evaluation. We use four processors, which reduces the forward model's runtime to roughly 4 seconds of computation. Thus our parallel MCMC scheme actually employs two levels of parallelism: an outer level involving parallel chains, as described in Section 3, and an inner level within each forward model evaluation.

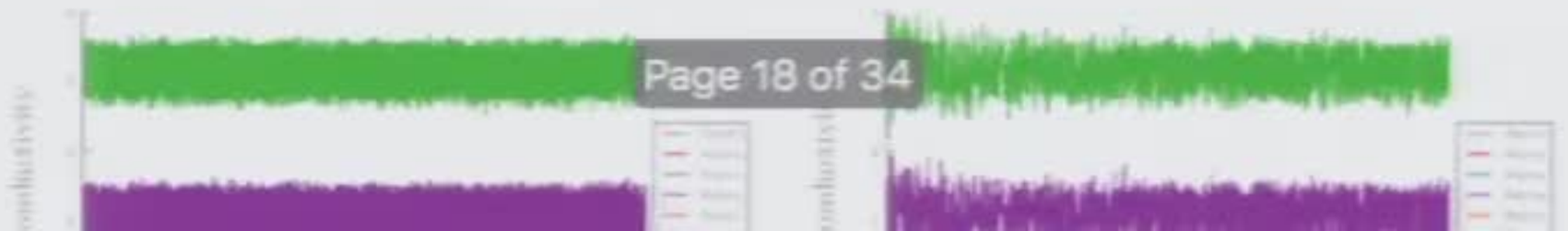


Figure 7. *The tracer concentration $c(x, y, t = 0.4)$, given the conductivity field in Figure 5. The tracer is injected from a well in each corner.*

SIAM/ASA J. UNCERTAINTY QUANTIFICATION
Vol. xx, pp. x

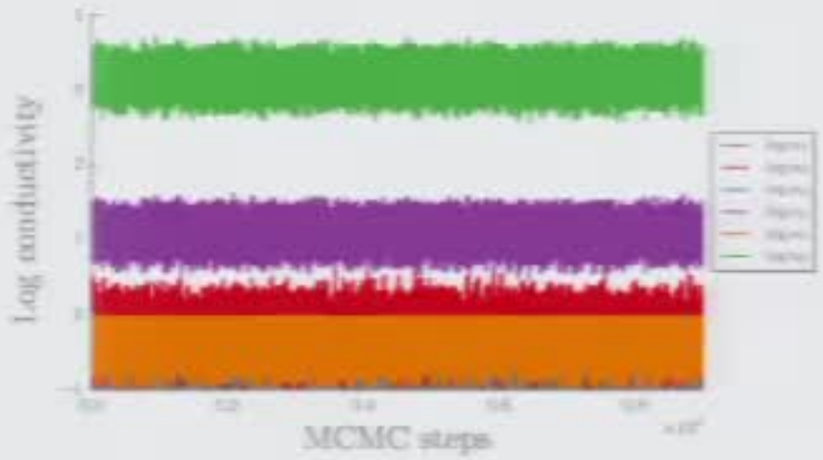
local approximations, we also wish to compare our approach with chains that employ exact evaluations of the forward model. To make such comparisons feasible—and also to reflect computational practice for complex PDE models—we parallelize each forward model evaluation. We use four processors, which reduces the forward model's runtime to roughly 4 seconds of computation. Thus our parallel MCMC scheme actually employs two levels of parallelism: an outer level involving parallel chains, as described in Section 3, and an inner level within each forward model evaluation.

(a) exact+AM (b) LA+AM

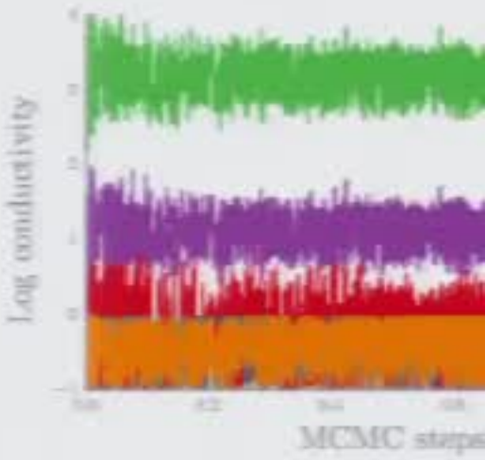
Figure 7. *The tracer concentration $c(x, y, t = 0.4)$, given the conductivity field is injected from a well in each corner.*

SIAM/ASA J. UNCERTAINTY QUANTIFICATION
Vol. xx, pp. x

local approximations, we also wish to compare our approach with chain
evaluations of the forward model. To make such comparisons feasible
computational practice for complex PDE models—we parallelize each
uation. We use four processors, which reduces the forward model's ru
seconds of computation. Thus our parallel MCMC scheme actually en
parallelism: an outer level involving parallel chains, as described in Sec
level within each forward model evaluation.



(a) exact+AM



(b) LA+A

Local polynomial surrogates

- Given n (potentially noisy) density evaluations at points $\{x^{(i)}\}_{i=1}^n$
- Find the degree p polynomial $\hat{\pi}(x)$ that minimizes the weighted least squares error

$$\hat{\pi}(x) = \arg\min_{\rho \in \mathcal{P}_p} \sum_{i=1}^{n} \left(\rho(x^{(i)}) - \pi(x^{(i)})\right)^2 K(x^{(i)}, x)$$

  - Locally supported kernel $K(\cdot, x)$

Local polynomial surrogates

- Given n (potentially noisy) density evaluations at points $\{x^{(i)}\}_{i=1}^n$
- Find the degree p polynomial $\hat{\pi}(x)$ that minimizes the weighted least squares error

$$\hat{\pi}(x) = \arg\min_{\rho \in \mathcal{P}_p} \sum_{i=1}^{n} \left(\rho(x^{(i)}) - \pi(x^{(i)})\right)^2 K(x^{(i)}, x)$$

  - Locally supported kernel $K(\cdot, x)$

- Known analytic derivatives
- Easily refined within MCMC (Conrad et al. JASA 2016

Local polynomial surrogates

- Given n (potentially noisy) density evaluations at points $\{x^{(i)}\}_{i=1}^n$
- Find the degree p polynomial $\hat{\pi}(x)$ that minimizes the weighted least squares error

$$\hat{\pi}(x) = \arg\min_{\rho \in \mathcal{P}_p} \sum_{i=1}^{n} \left(\rho(x^{(i)}) - \pi(x^{(i)})\right)^2 K(x^{(i)}, x)$$

- Locally supported kernel $K(\cdot, x)$

Local polynomial surrogates

- Given n (potentially noisy) density evaluations at points $\{x^{(i)}\}_{i=1}^n$
- Find the degree p polynomial $\hat{\pi}(x)$ that minimizes the weighted least squares error

Bayesian computation in hierarchical models using marginal local approximation MCMC

[1]**Andrew D. Davis**, [1]Youssef Marzouk, [2]Aaron Smith, [3]Natesh Pillai

[1]Department of Aeronautics and Astronautics
Center for Computational Engineering
Massachusetts Institute of Technology
http://uqgroup.mit.edu

[2]Department of Mathematics and Statistics
University of Ottawa

[3]Department of Statistics
Harvard University

Bayesian computation in hierarchical models using marginal local approximation MCMC

[1]**Andrew D. Davis**, [1]Youssef Marzouk, [2]Aaron Smith, [3]Natesh Pillai

[1]Department of Aeronautics and Astronautics
Center for Computational Engineering
Massachusetts Institute of Technology
http://uqgroup.mit.edu

[2]Department of Mathematics and Statistics
University of Ottawa

[3]Department of Statistics
Harvard University

April 2018

Problem statement

We want to **characterize the distribution π** using a sampling method

Problem statement

We want to **characterize the distribution π** using a sampling method

Two problems:

1. The probability density function $\pi(x)$ is **computationally expensive**

Problem statement

We want to characterize the distribution π using a sampling method

Two problems:

1. The probability density function $\pi(x)$ is computationally expensive

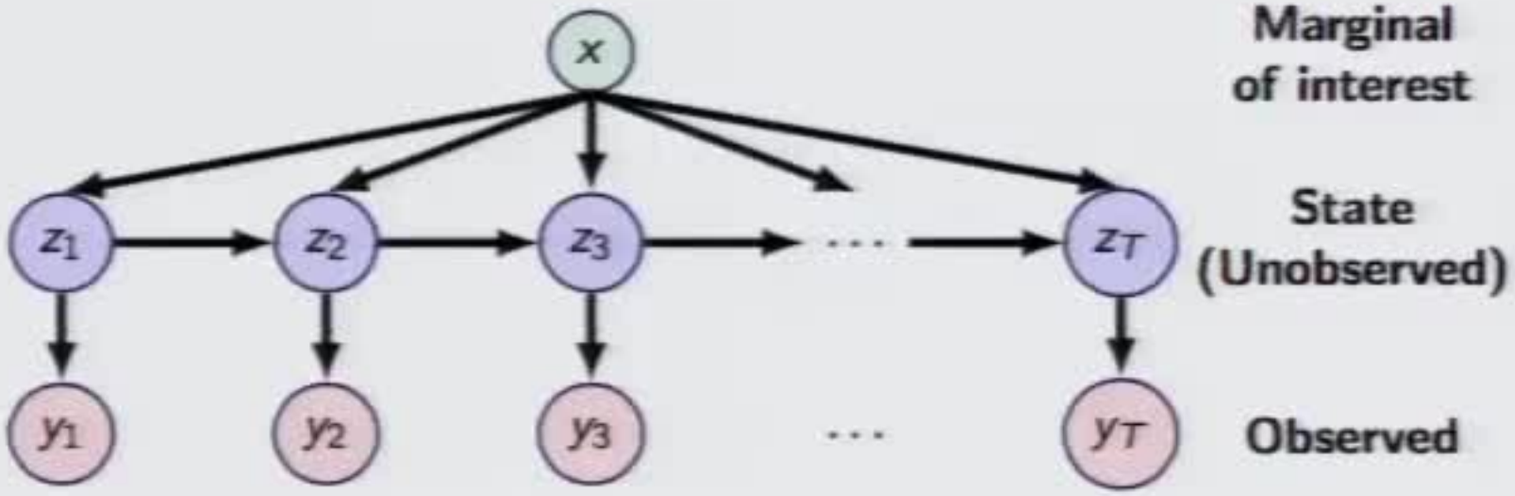
Page 3 of 69

2. Only **noisy** density evaluations are available $\tilde{\pi}(x) = \pi(x) + \epsilon$

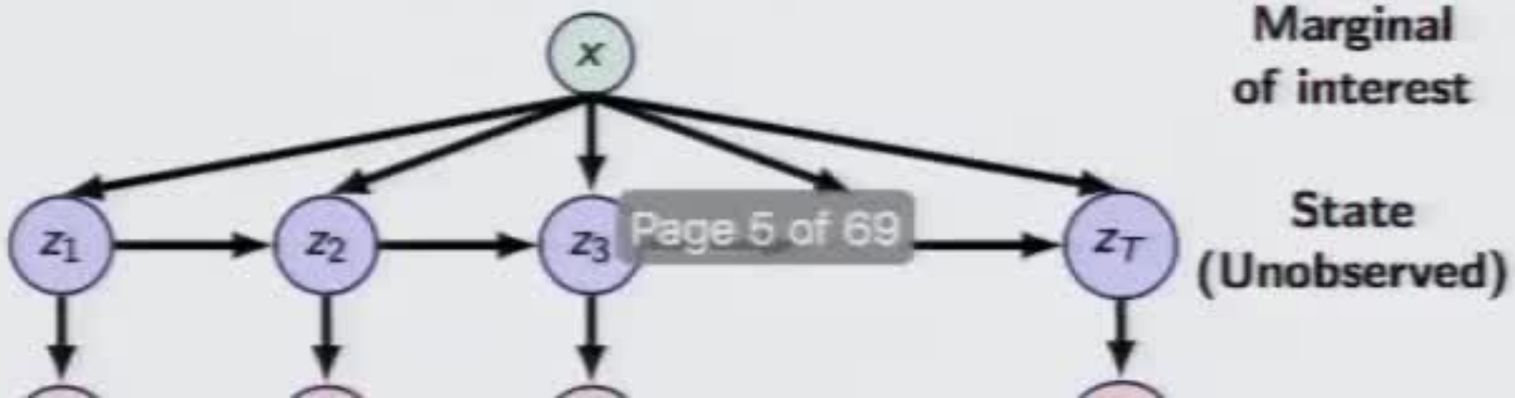
Motivating example: state space modelling

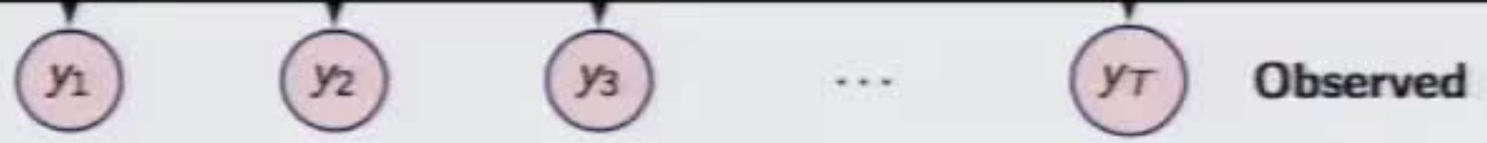
Static parameter estimation—

- Given: data $y_{1:T}$ with an unobservable state $z_{1:T}$
- We want to characterize the distribution over $x|y_{1:T}$
  - x is low-dimensional



Example: state space modelling





Marginalizing avoids characterizing the joint density over high dimensional parameters $[x, z_{1:T}]$

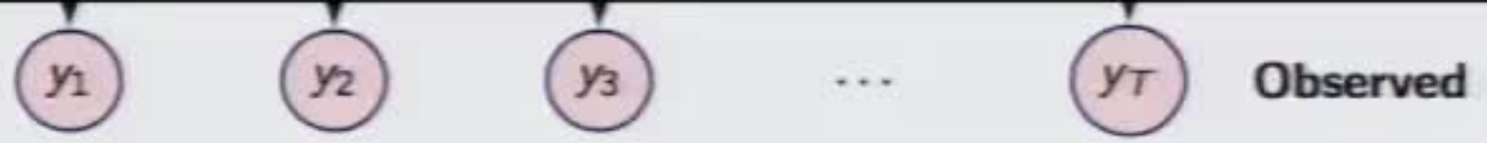
$$\underbrace{\pi(x|y_{1:T})}_{\text{Posterior}} \propto \underbrace{\pi(x)}_{\text{Prior}} \underbrace{\int_{\mathcal{Z}} \pi(z_{1:T}, y_{1:T}|x)\, dz}_{\text{Likelihood}}$$

Example: state space modelling

$$\underbrace{\pi(x|y_{1:T})}_{\text{Posterior}} \propto \underbrace{\pi(x)}_{\text{Prior}} \underbrace{\int_{\mathcal{Z}} \pi(z_{1:T}, y_{1:T}|x)\, dz}_{\text{Likelihood}}$$

We only have a **noisy estimate of the likelihood**

$$\int_{\mathcal{Z}} \pi(z_{1:T}, y_{1:T}|x)\, dz \approx \sum_{i=1}^{N} w^{(i)} \pi(z_{1:T}^{(i)}, y_{1:T}|x)$$



Marginalizing avoids characterizing the joint density over high dimensional parameters $[x, z_{1:T}]$

$$\underbrace{\pi(x|y_{1:T})}_{\text{Posterior}} \propto \underbrace{\pi(x)}_{\text{Prior}} \underbrace{\int_{\mathcal{Z}} \pi(z_{1:T}, y_{1:T}|x)\, dz}_{\text{Likelihood}}$$

Example: state space modelling

$$\underbrace{\pi(x|y_{1:T})}_{\text{Posterior}} \propto \underbrace{\pi(x)}_{\text{Prior}} \underbrace{\int_{\mathcal{Z}} \pi(z_{1:T}, y_{1:T}|x)\, dz}_{\text{Likelihood}}$$

We only have a **noisy estimate of the likelihood**

$$\int_{\mathcal{Z}} \pi(z_{1:T}, y_{1:T}|x)\, dz \approx \sum_{i=1}^{N} w^{(i)} \pi(z_{1:T}^{(i)}, y_{1:T}|x)$$

Example: state space modelling

$$\underbrace{\pi(x|y_{1:T})}_{\text{Posterior}} \propto \underbrace{\pi(x)}_{\text{Prior}} \underbrace{\int_{\mathcal{Z}} \pi(z_{1:T}, y_{1:T}|x)\, dz}_{\text{Likelihood}}$$

We only have a **noisy estimate of the likelihood**

$$\int_{\mathcal{Z}} \pi(z_{1:T}, y_{1:T}|x)\, dz \approx \sum_{i=1}^{N} w^{(i)} \pi(z_{1:T}^{(i)}, y_{1:T}|x)$$

- Parameter space is partitioned into coordinates characterized by MCMC (x) and coordinates to be "marginalized away" ($z_{1:T}$)
- Model evaluations are (even more) computationally expensive
- We only have noisy target density evaluations

Approaches & outline

- Pseudo-marginal MCMC can characterize the posterior marginal distribution (Beaumont, 2010) and (Andrieu and Roberts, 2009)
  - Computationally infeasible!

- We only have noisy target density evaluations

Approaches & outline

- Pseudo-marginal MCMC can characterize the posterior marginal distribution (Beaumont, 2010) and (Andrieu and Roberts, 2009)
  - Computationally infeasible!

Approaches & outline

- Pseudo-marginal MCMC can characterize the posterior marginal distribution (Beaumont, 2010) and (Andrieu and Roberts, 2009)
  - Computationally infeasible!

- We only have noisy target density evaluations

Approaches & outline

- Pseudo-marginal MCMC can characterize the posterior marginal distribution (Beaumont, 2010) and (Andrieu and Roberts, 2009)
  - Computationally infeasible!

Approaches & outline

- Pseudo-marginal MCMC can characterize the posterior marginal distribution (Beaumont, 2010) and (Andrieu and Roberts, 2009)
  - Computationally infeasible!

- Ideally, we balance MCMC variance with surrogate bias squared to derive an ideal refinement rate

$$C_{MCMC} T^{-1} \sim C^2_{Surrogate} k \bar{\lambda}^2 \Delta^{2(p+1)}$$

Structural refinement strategy

- We devise a refinement strategy based on a local error estimate to balance MCMC variance and structural error

Structural refinement strategy

Page 44 of 69

- We devise a refinement strategy based on a local error estimate to balance MCMC variance and structural error

- Ideally, we balance MCMC variance with surrogate bias squared to derive an ideal refinement rate

$$C_{MCMC} T^{-1} \sim C^2_{Surrogate} k \bar{\Lambda}^2 \Delta^{2(p+1)}$$

Structural refinement strategy

- We devise a refinement strategy based on a local error estimate to balance MCMC variance and structural error

Structural refinement strategy

Page 44 of 69

- We devise a refinement strategy based on a local error estimate to balance MCMC variance and structural error

Structural refinement strategy

- We devise a refinement strategy based on a local error estimate to balance MCMC variance and structural error
- Divide the chain into M levels and prescribe an error threshold $\gamma(M) = \gamma_0 M^{-\gamma_1}$ on each level
  - Explore the parameter space before refining the error threshold
- We switch to level $M+1$ at step T_M, when the MCMC variance balances the structural error

$$\underbrace{C_{MCMC} T_M^{-1}}_{\text{MCMC variance}} = \underbrace{\gamma_0^2 M^{-2\gamma_1}}_{[\text{Structural bias}]^2}$$

$$T_M = \varphi_1 M^{2\gamma_1}$$

Davis et al. MIT 21 / 32

Structural refinement strategy

- We devise a refinement strategy based on a local error estimate to balance MCMC variance and structural error
- Divide the chain into M Page 48 of 69 ribe an error threshold $\gamma(M) = \gamma_0 M^{-\gamma_1}$ on each level
  - Explore the parameter space before refining the error threshold

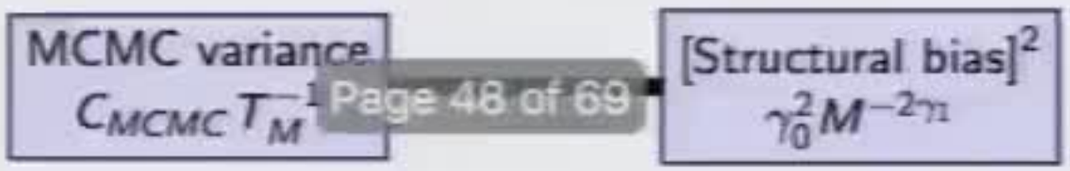
balance MCMC variance and structural error

- Divide the chain into M levels and prescribe an error threshold $\gamma(M) = \gamma_0 M^{-\gamma_1}$ on each level
  - Explore the parameter space before refining the error threshold
- We switch to level $M+1$ at step T_M, when the MCMC variance balances the structural error

$$T_M = \varphi_1 M^{2\gamma_1}$$

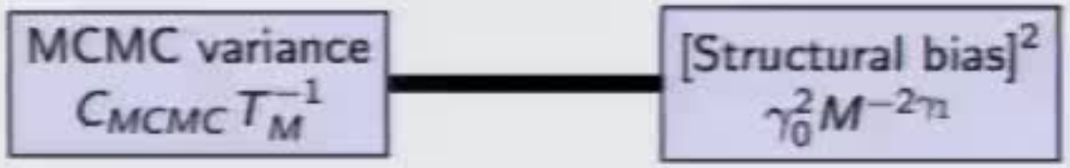
Structural refinement strategy

- We devise a refinement strategy based on a local error estimate to balance MCMC variance and structural error
- Divide the chain into M levels and prescribe an error threshold $\gamma(M) = \gamma_0 M^{-\gamma_1}$ on each level
  - Explore the parameter space before refining the error threshold
- We switch to level $M+1$ at step T_M, when the MCMC variance balances the structural error



$$T_M = \varphi_1 M^{2\gamma_1}$$

- We switch to level $M+1$ at step T_M, when the MCMC variance balances the structural error



$$T_M = \varphi_1 M^{2\gamma_1}$$

- The threshold decay rate must be $\gamma_1 > 0.5$ so the length of each level $T_M - T_{M-1}$ grows

We trigger refinement based on a piecewise constant error threshold

Structural refinement

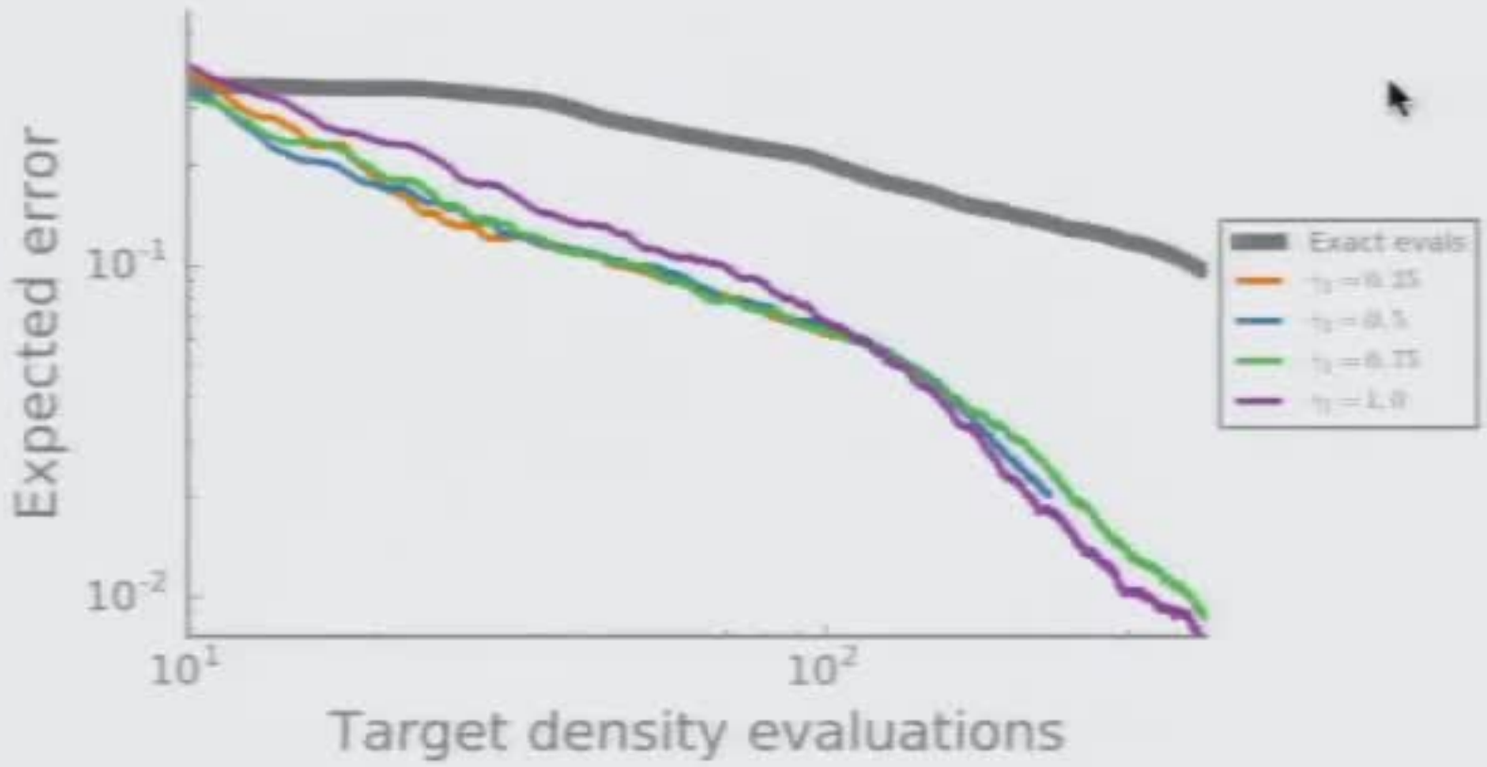
At $x^{(t)}$, refine the surrogate if:

1. The poisedness constant is too large $\Lambda^{(t)} > \bar{\Lambda}$
2. The local error indicator is greater than the level's threshold

$$e(x^{(t)}) = \sqrt{k}\Lambda(x^{(t)})\Delta^{p+1}(x^{(t)}) > \gamma_0 M^{-\gamma_1}$$

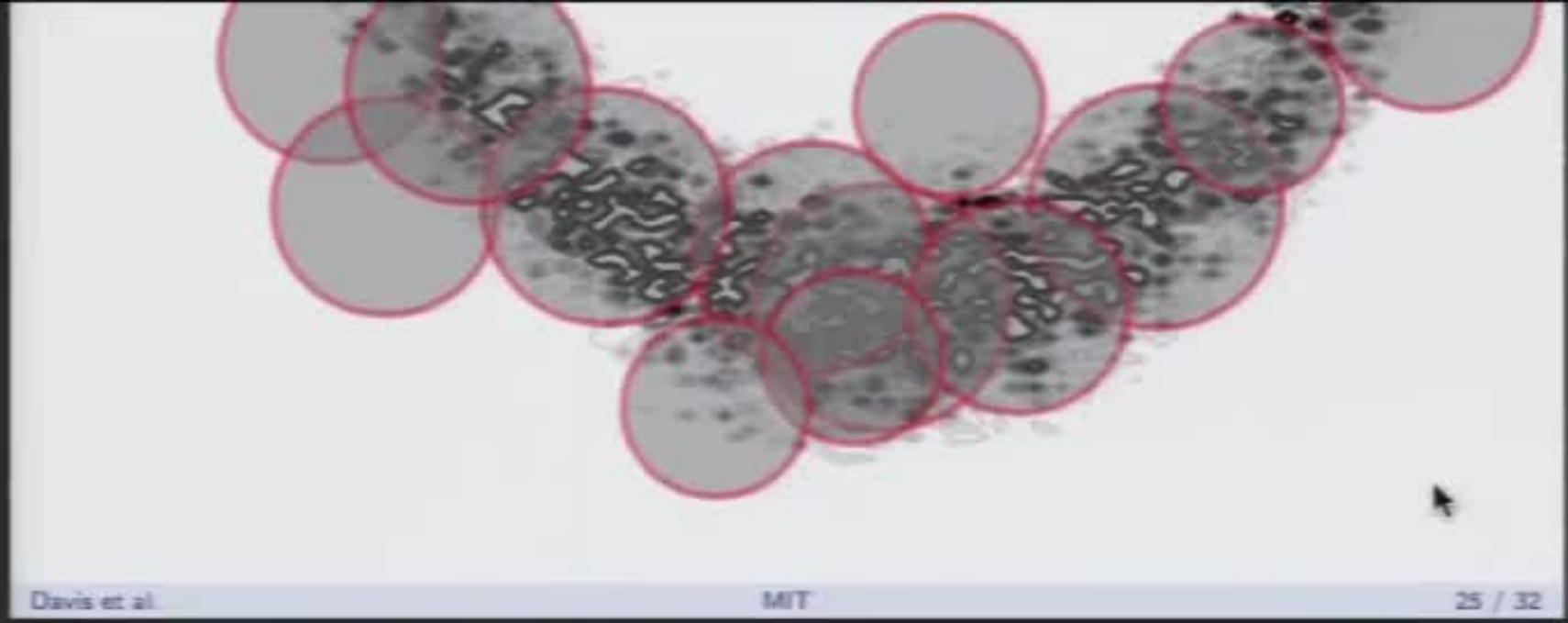
Expected error (structural refinement)

- The expected number of refinements is the same (when $\gamma_1 > 0.5$)
- When $\gamma_1 \leq 0.5$, the surrogate is underrefined
  - The error is dominated by the structural bias



Local polynomial surrogates

- Now consider situations with **noisy evaluations** of the target density



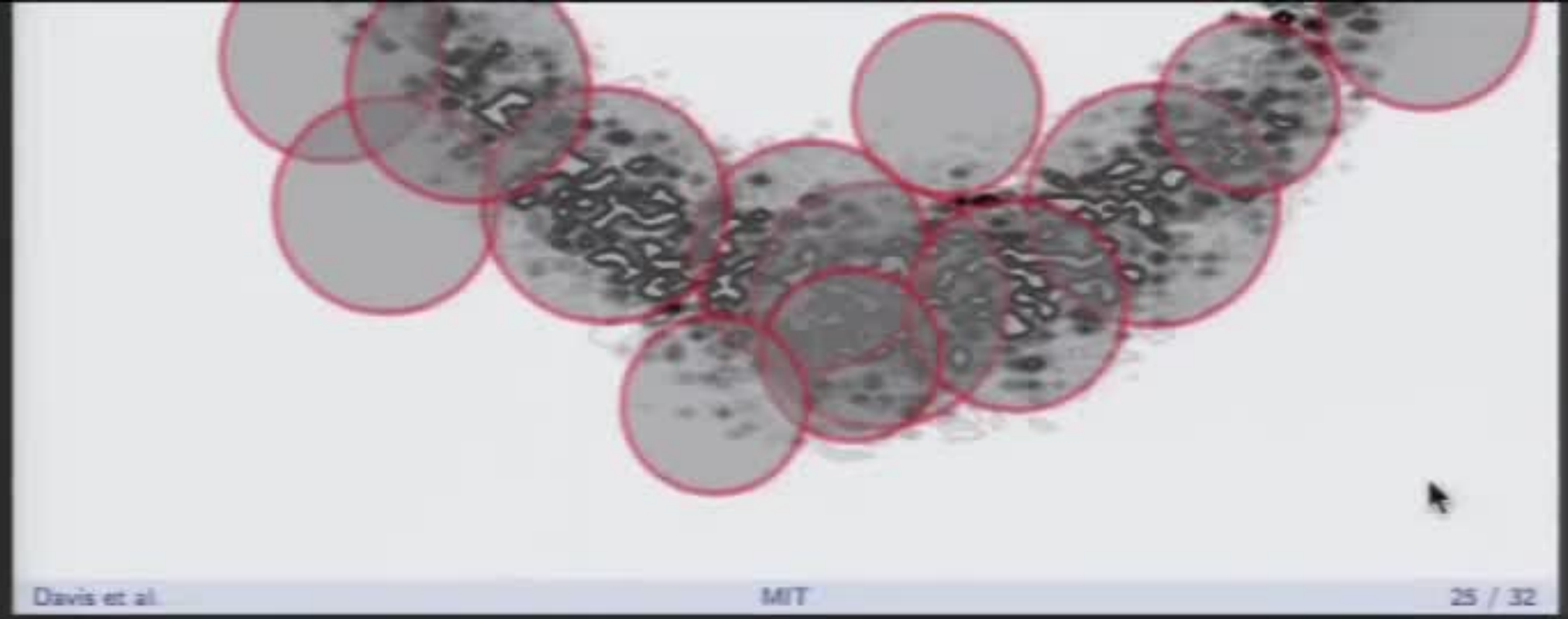
Surrogate convergence (noisy density evaluations)

In the **noisy evaluation** case, the surrogate $\hat{\pi}(x)$ approaches $\pi(x)$ as:

1. The ball size $\Delta \to 0$ (number of evaluations $n \to \infty$)
2. Λ-poisedness is maintained inside each ball
3. The number of nearest neighbors $k \to \infty$ (while $\frac{k}{n} \to 0$)

Large balls

Small balls



Surrogate convergence (noisy density evaluations)

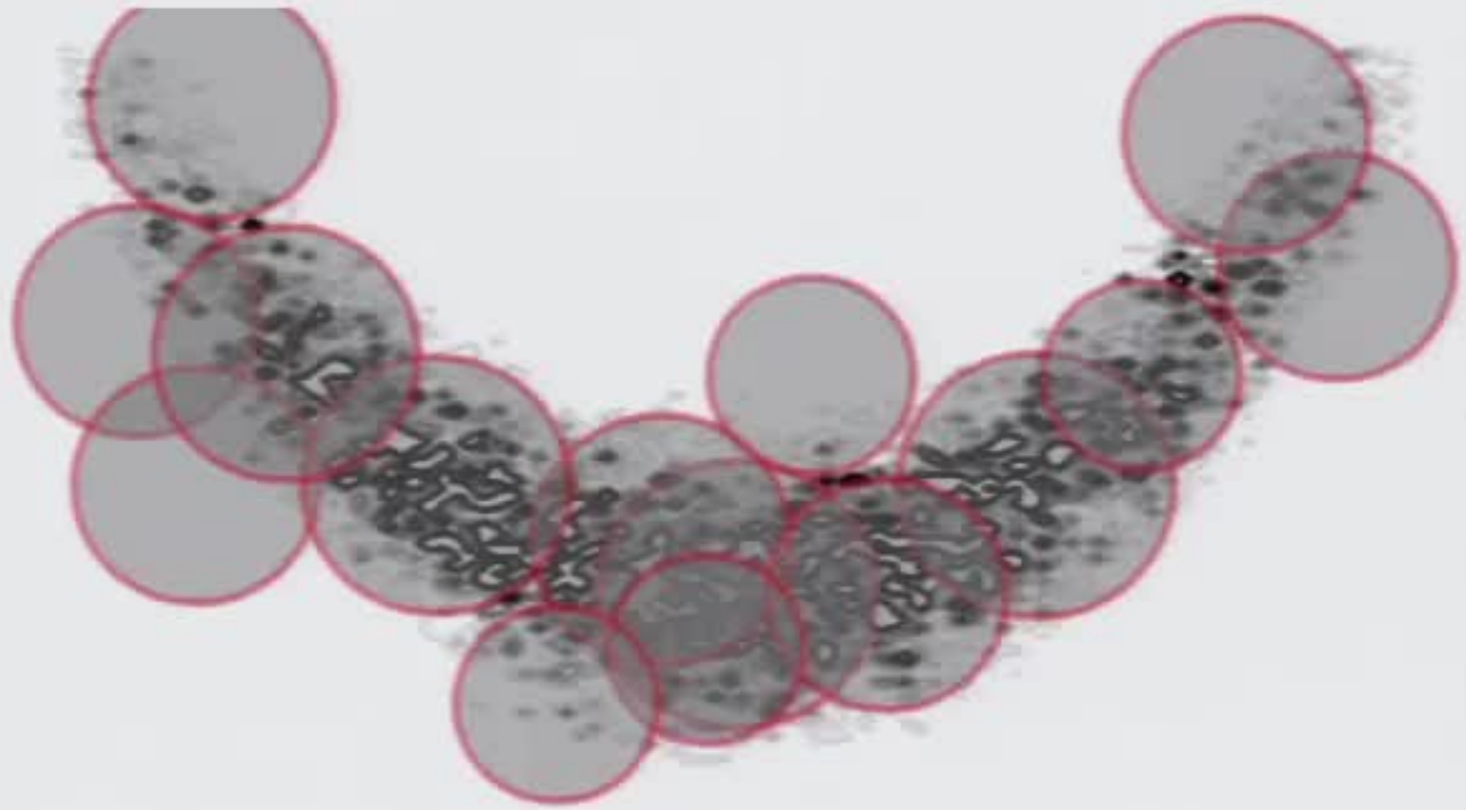
In the **noisy evaluation** case, the surrogate $\hat{\pi}(x)$ approaches $\pi(x)$ as:

1. The ball size $\Delta \to 0$ (number of evaluations $n \to \infty$)
2. Λ-poisedness is maintained inside each ball
3. The number of nearest neighbors $k \to \infty$ (while $\frac{k}{n} \to 0$)

Large balls **Small balls**

Local polynomial surrogates

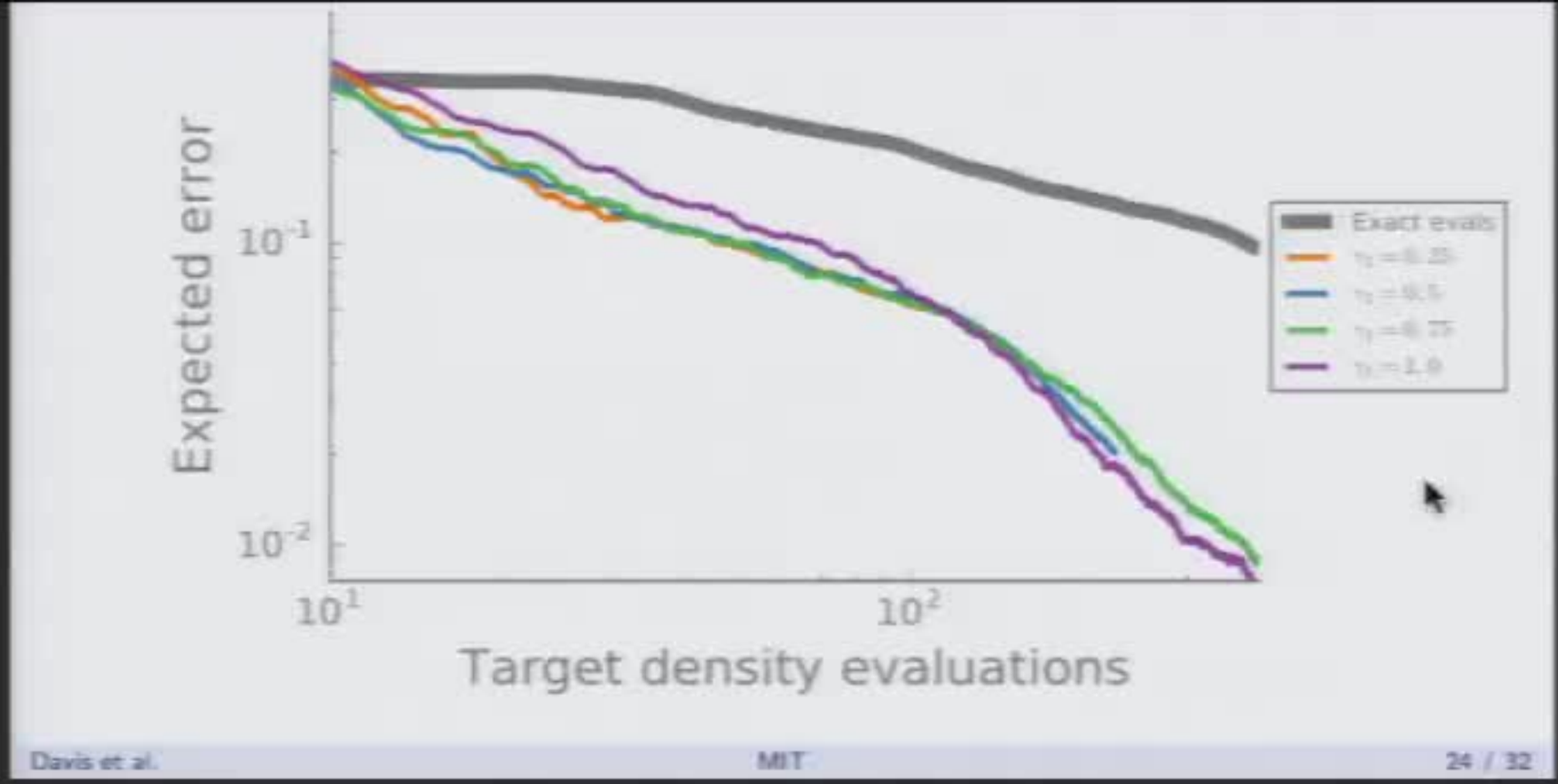
- Now consider situations with **noisy evaluations** of the target density



Surrogate convergence (noisy density evaluations)

In the **noisy evaluation** case, the surrogate $\hat{\pi}(x)$ approaches $\pi(x)$ as:

1. The ball size $\Delta \to 0$ (number of evaluations $n \to \infty$)



Local polynomial surrogates

- Now consider situations with **noisy evaluations** of the target density