Adaptive Posterior Approximation within MCMC

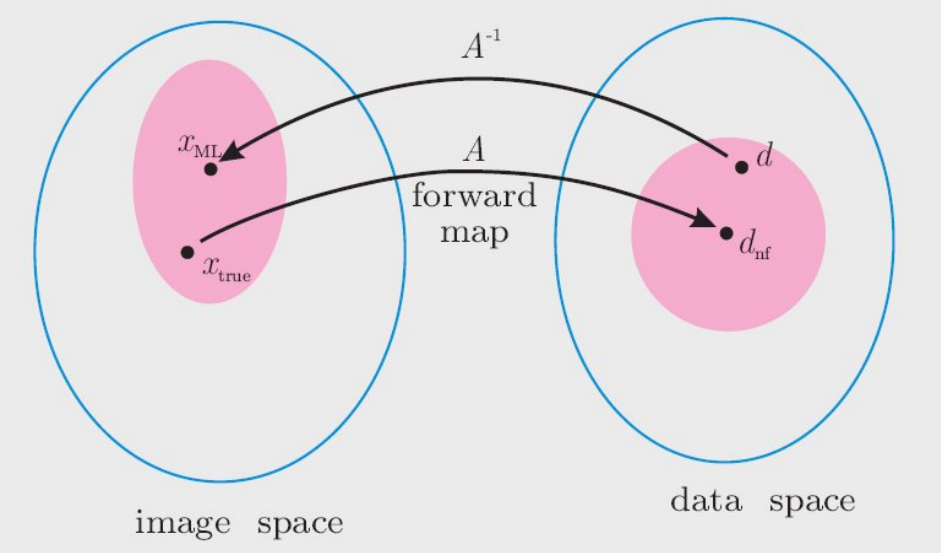
Tiangang Cui (MIT)
Colin Fox (University of Otago)
Mike O'Sullivan (University of Auckland)
Youssef Marzouk (MIT)
Karen Willcox (MIT)

06/January/2012

Outline

Bayesian framework: Inverse Problems

$d = A(x) + e$: data d, image x, measurement noise e, forward map A



Bayes' rule

$$\pi(x, \alpha, \beta|d) \propto L(d|x, \alpha)\, \pi(x|\beta)\, \pi(\alpha)\pi(\beta)$$

Construction of the Posterior Distribution

- Likelihood: knowledge of measurement noise, maybe knowledge of model error
- Prior: stochastic modelling of x, physical laws, previous data, expert opinion

Bayesian framework: Inverse Problems

Prior

- Expert knowledge, e.g. porosity is between 0 and 1 and more likely < 0.5, and permeability range is high, low, or medium for some specific regions. (Good for physical bounds)
- Spatial statistics: e.g. Gaussian Markov Random field (GMRF) or Kriging/ Gaussian process

Likelihood

Suppose that the model noise $n \sim f_\alpha(\cdot)$

$$L(d|x,\alpha) = f_\alpha\left(d - A(x)\right)$$

We assume the noise follows i.i.d. Gaussian, $\alpha = \Sigma_e$

$$L\left(d|x\right) \propto \exp\left(-\frac{1}{2}(d - A(x))^T \Sigma_n^{-1}(d - A(x))\right)$$

Bayesian framework: Solution Approaches

Optimization (MAP)

maximize the posterior density (in a logarithmic scale)

$$\text{argmax}_{x \in \mathcal{X}} \left[-\frac{1}{2}(d - A(x))^T \Sigma_n^{-1} (d - A(x)) + \log(\pi(x)) \right]$$

Calculating Expectation

Summarize information over the posterior distribution by calculating the expected value of function of interest

$$\mathrm{E}_\pi [f(x)] = \int_X f(x)\, \pi(x|d)\, dx$$

Example: mean $\mathrm{E}_\pi [x]$, covariance $\mathrm{Var}_\pi [x]$, mean pressure prediction $\mathrm{E}_\pi [p(A(x))]$, covariance of pressure prediction $\mathrm{Var}_\pi [p(A(x))]$ ...

- High-dimensional integrals $\Rightarrow$ Monte Carlo integration

$$x^{(1)}, \ldots, x^{(n)} \sim \pi(\cdot|d) \qquad \mathrm{E}_\pi [f(x)] \approx \frac{1}{n} \sum_{i=1}^n f\left(x^{(i)}\right)$$

Posterior Sampling – Metropolis Hastings

Monte Carlo integration needs samples from $\pi(x|d)$, MH is the most widely used

1. given state $X_t = x$ at time t generate candidate state y from a proposal distribution $q(x, \cdot)$
2. With probability $\alpha(x, y) = \min\left(1, \frac{\pi(y|d)q(y,x)}{\pi(x|d)q(x,y)}\right)$ set $X_{t+1} = y$ otherwise $X_{t+1} = x$
3. Repeat

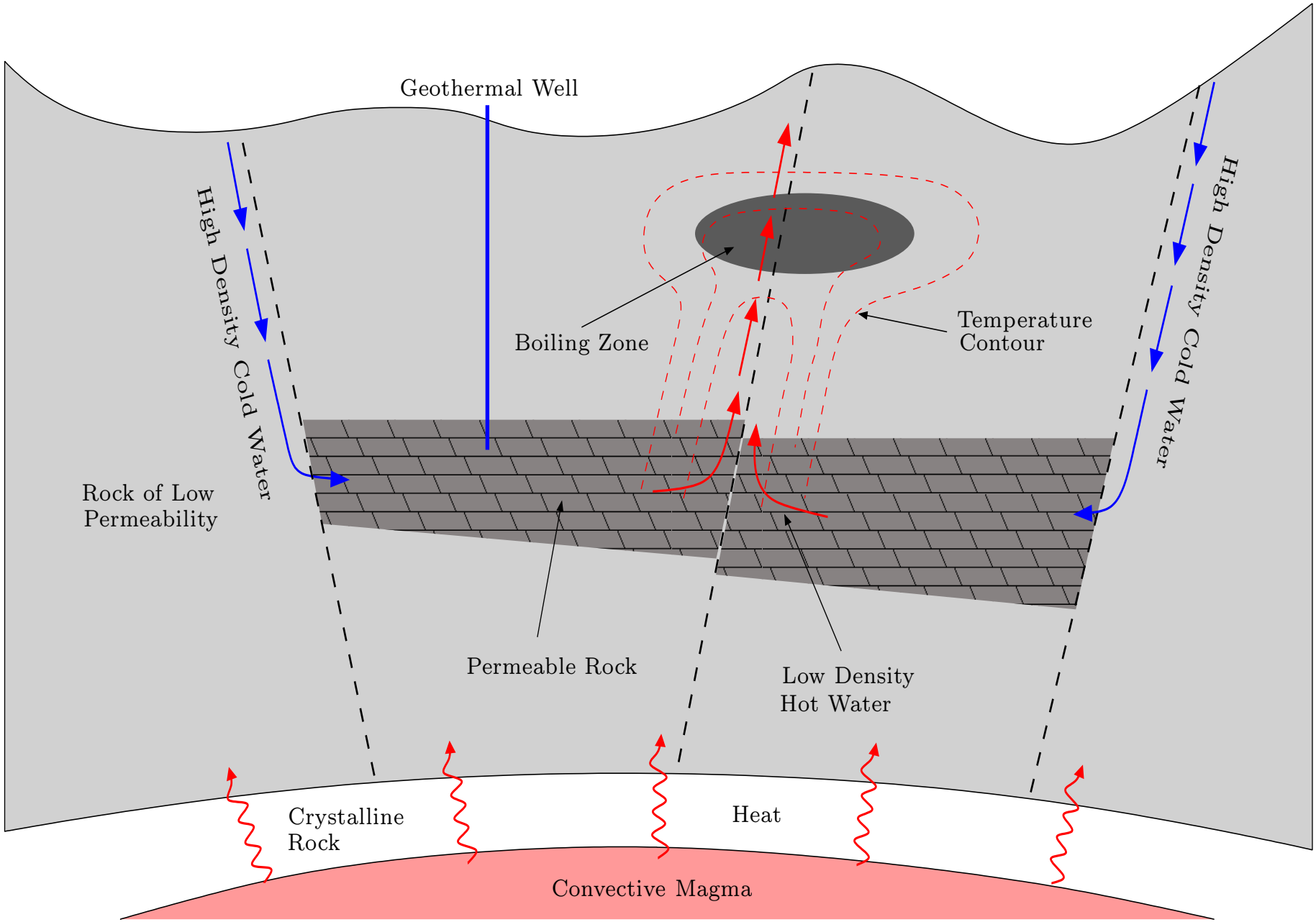
Remarks

- The proposal $q(x, \cdot)$ defines how the algorithm traverse the parameter space, and hence the efficiency. Many works in statistics and physics, Adaptive Metropolis, MALA, Stochastic Newton, Manifold MCMC, Particle MCMC ...
- In our work, we built **fast approximation** to the posterior distribution. So we can draw more samples in a given budget of CPU time.
- Two different methods will be discussed.

Outline

Example: Geothermal Reservoir Modelling

Example: Geothermal Reservoir Modelling



Example: Governing Equations

Multiphase non-isothermal flow are governed by mass and energy balance equations

$$\frac{d}{dt}\int_\Omega M\,dV = \int_{\partial\Omega} Q\cdot\hat{n}\,d\Gamma + \int_\Omega q dV$$

and multiphase Darcy's law

$$M_{\mathrm{m}} = \phi(\rho_{\mathrm{l}} S_{\mathrm{l}} + \rho_{\mathrm{v}} S_{\mathrm{v}})$$

$$Q_{\mathrm{m}} = \sum_{\beta=\mathrm{l},\mathrm{v}} \frac{k k_{\mathrm{r}\beta}}{\nu_\beta}(\nabla p - \rho_\beta \hat{\mathrm{g}})$$

$$M_{\mathrm{e}} = (1-\phi)\rho_{\mathrm{r}} c_{\mathrm{r}} T + \phi(\rho_{\mathrm{l}} u_{\mathrm{l}} S_{\mathrm{l}} + \rho_{\mathrm{v}} u_{\mathrm{v}} S_{\mathrm{v}})$$

$$Q_{\mathrm{e}} = \sum_{\beta=\mathrm{l},\mathrm{v}} \frac{k k_{\mathrm{r}\beta}}{\nu_\beta}(\nabla p - \rho_\beta \hat{\mathrm{g}}) h_\beta - K\nabla T$$

- M – mass (m) or energy (e) per unit volume
- Q – mass (m) or energy (e) flux
- q – mass (m) or energy (e) input/output rate
- ϕ, k and $k_{\mathrm{r}\beta}$ are the porosity, permeability and relative permeability, respectively.
- The governing PDEs are solved by finite volume solver TOUGH2, Pruess (1991).

Example: Calibration

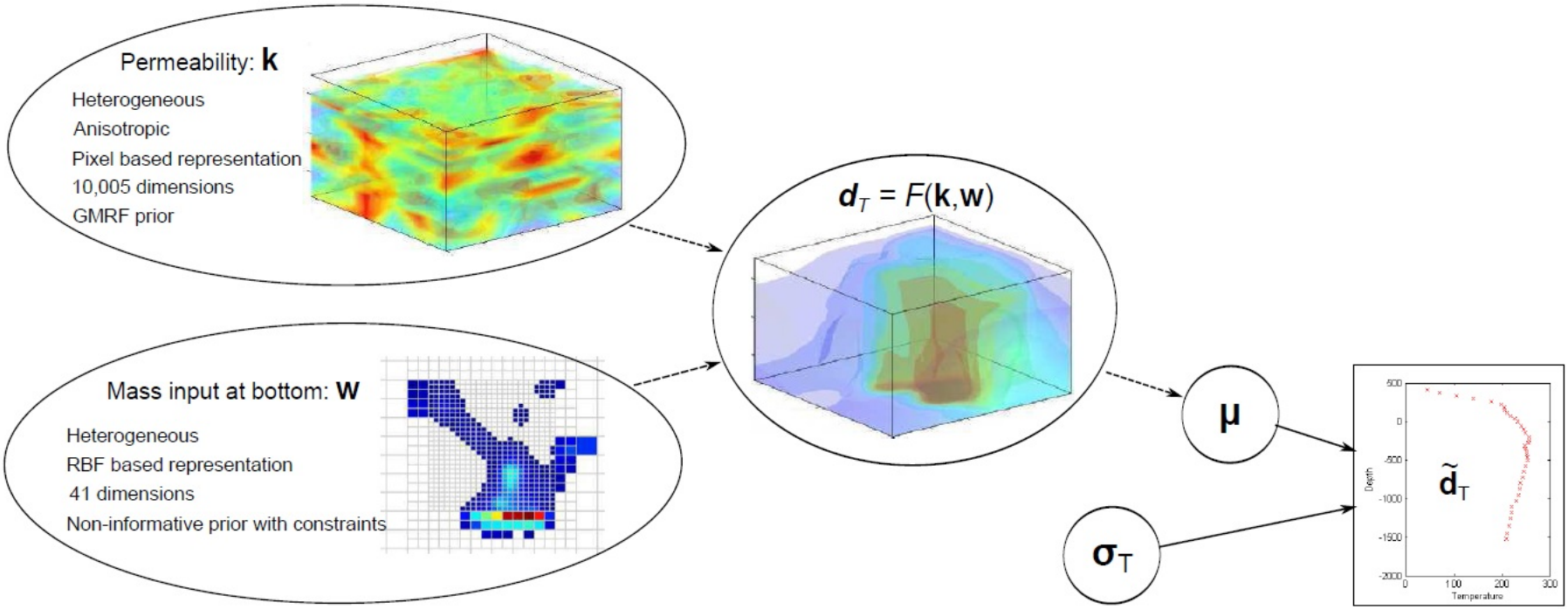
Wish to

- Estimate the values of unknowns (permeability, porosity, and boundary condition, etc.) from data (temperature, pressure, and enthaly, etc.)
- Predict the long term behavior of geothermal reservoirs (presure responses and enthalpy response under centain pump rates)
- Assess the uncertainty of this prediction, and hence the risk

Difficulties

- Models are computationally very demanding, from the order of hours to days.
- Parameter to data map (forward model) is highly nonlinear
- The data are sparsely measured and corrupted by noise
- Black box model – no adjoint gradient and Hessian

Case Study: Natural State Modeling



- Parameters: permeability field, and boundary condition. $\sim 10,000$
- Data: temperature measurements from wellbores. ($\sim$ 230 points)
- Computing time: Each model evaluation $\sim$ 30 - 50 mins

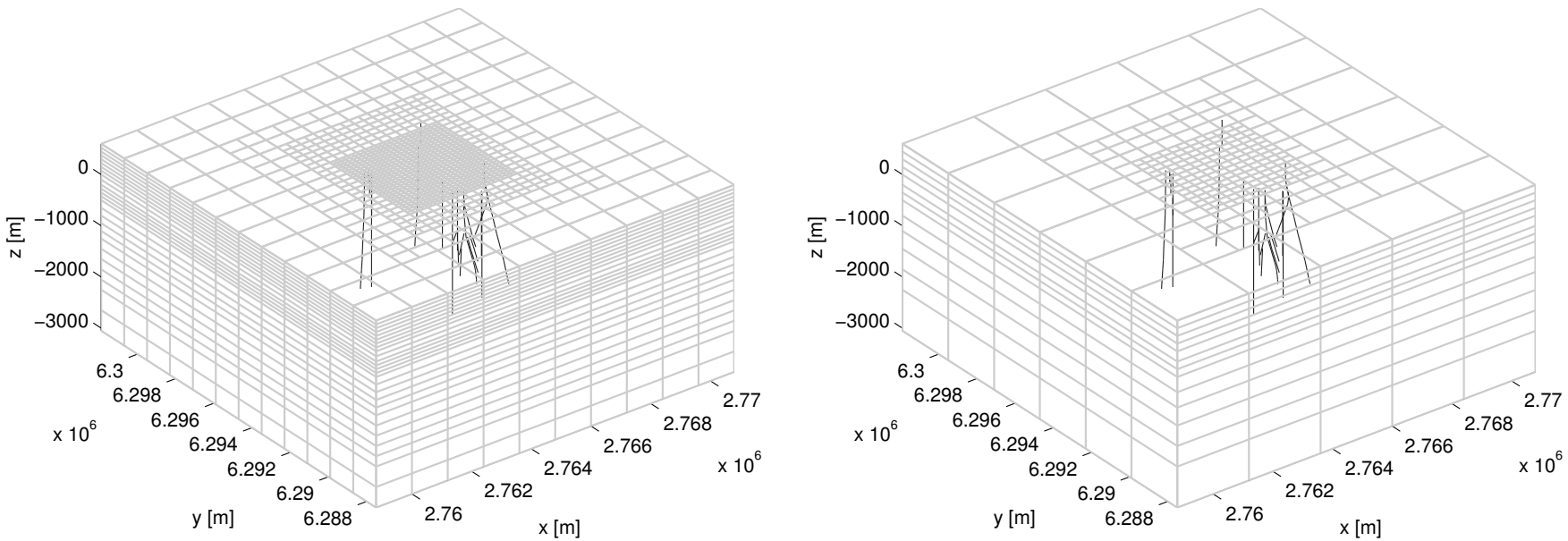
Using Approximation: Two step MCMC

Delayed Acceptance – Christen and Fox, 2005

1. Given state $X_t = x$, generate candidate state y from a proposal $q(x, \cdot)$
2. With probability
$$\alpha(x, y) = \min\left[1, \frac{\pi^*(y|d)}{\pi^*(x|d)} \frac{q(y, x)}{q(x, y)}\right],$$
accept y, then go to step 3. Otherwise reject y setting $X_{t+1} = x$, then go to step 4
3. With probability
$$\beta(x, y) = \min\left[1, \frac{\pi(y|d)}{\pi(x|d)} \frac{\pi^*(x|d)}{\pi^*(y|d)}\right],$$
accept y setting $X_{t+1} = y$. Otherwise reject y setting $X_{t+1} = x$.
4. repeat

What π^* we can use?

Using Approximation: Approximation Error Model



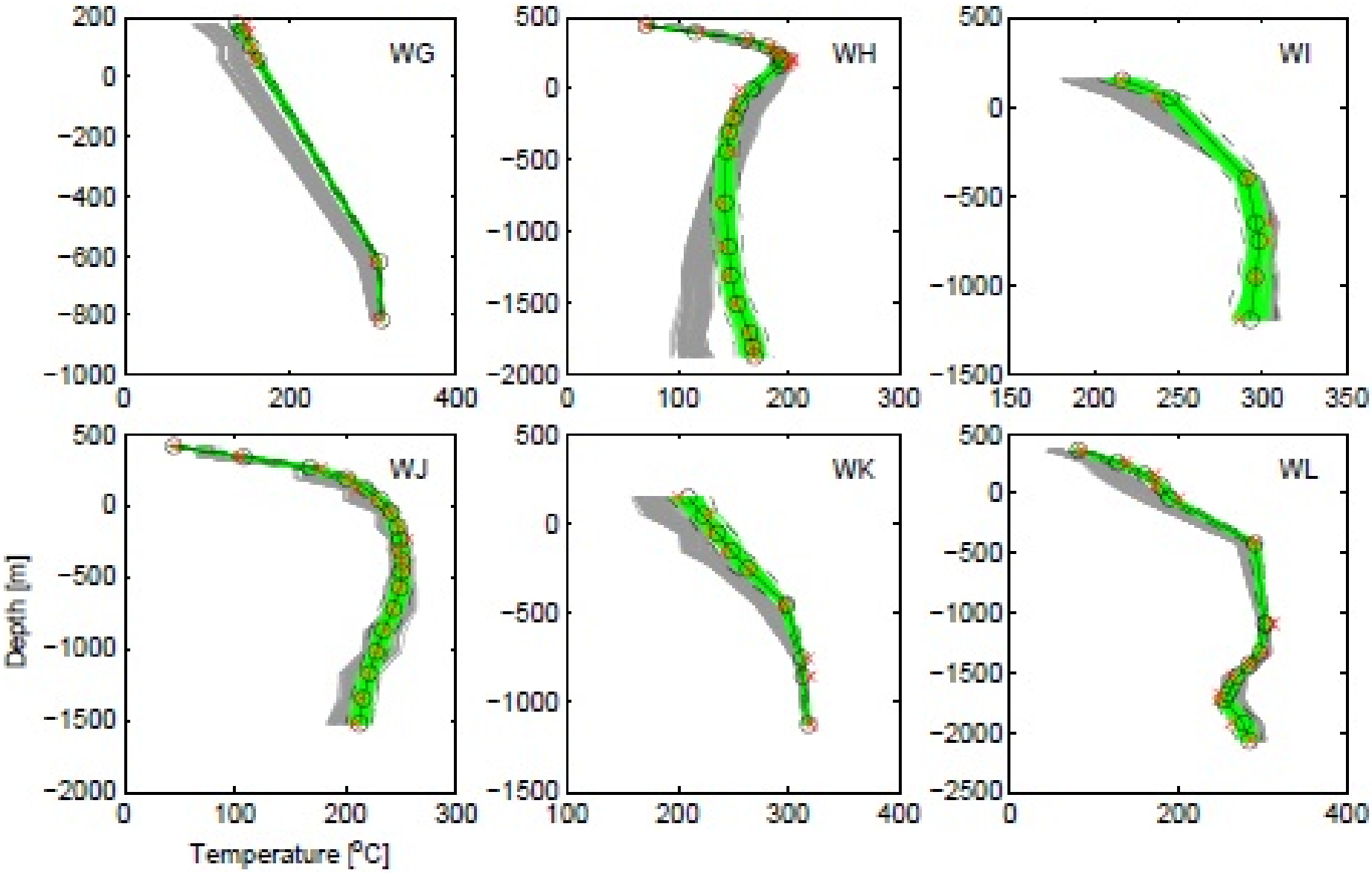
Computing time: fine $\sim$ 30 - 50 mins, coarse $\sim$ 1 mins

Version 1

Replace the fine model $A(x)$ with the coarse model $A^*(x)$ in the likelihood function:

$$L^*(d|x) \propto \exp\left\{-\frac{1}{2}[A^*(x)-d]^T\Sigma_e^{-1}[A^*(x)-d]\right\}$$

Using Approximation: Approximation Error Model



Using Approximation: Approximation Error Model

Version 2

To deal the model reduction error, Kaipio and Somersalo (2007) introduced an approximation error model:

Revise the parameter to data map

$$\begin{aligned} d &= A(x) + e \\ &= A^*(x) + [A(x) - A^*(x)] + e \\ &= A^*(x) + B(x) + e \end{aligned}$$

Kaipio and Somersalo (2007) assume $B \perp x$ and $B \sim N(\mu_B, \Sigma_B)$, yields the approximate likelihood

$$L^*(d|x) \propto \exp\left\{ -\frac{1}{2}[A^*(x) + \mu_B - d]^T (\Sigma_B + \Sigma_e)^{-1} [A^*(x) + \mu_B - d] \right\}$$

Using Approximation: A Priori vs. A Posteriori

A Priori – Kaipio and Somersalo, 2007

- pre-sample a set of parameters $x_1, \ldots, x_N$ from $\pi(x)$ – very fast
- evaluate the model reduction error $A(x_i) - A^*(x_i)$ for each x_i – slow
- use Monte Carlo integration to calculate μ_B and Σ_B

This yields an estimation over the **prior** distribution

$$\begin{aligned} \mu_B &= \int_{\mathcal{X}} B(x)\pi(x)dx, \\ \Sigma_B &= \int_{\mathcal{X}} [B(x) - \mu_B][B(x) - \mu_B]^T \pi(x)dx, \end{aligned}$$

A Posteriori – Cui, Fox, and O'Sullivan, 2010

By integrating the approximation error model into an adaptive two step MCMC, we provide an **online** estimation over the **posterior** distribution:

$$\begin{aligned} \mu_B &= \int_{\mathcal{X}} B(x)\pi(x|d)dx, \\ \Sigma_B &= \int_{\mathcal{X}} [B(x) - \mu_B][B(x) - \mu_B]^T \pi(x|d)dx, \end{aligned}$$

Using Approximation: Adaptive Delayed Acceptance

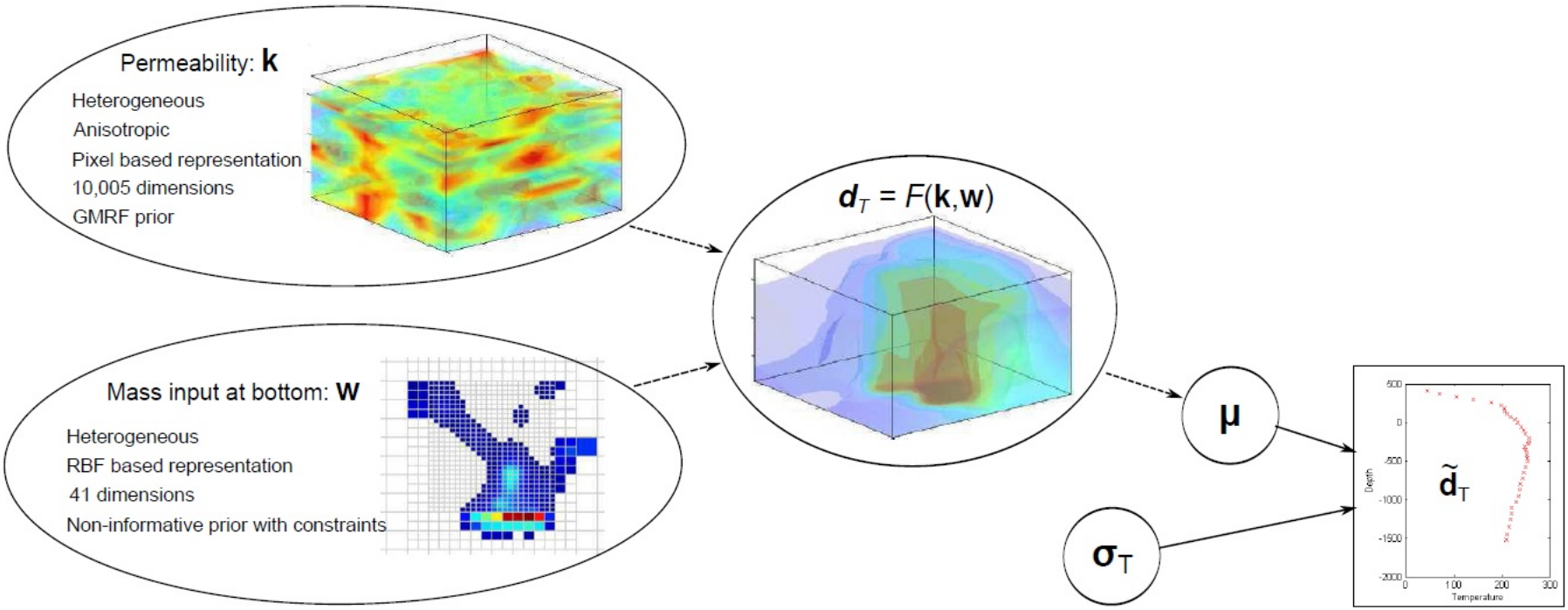
Adaptive Delayed Acceptance (ADA)

1. Given state $X_t = x$, generate candidate state y from a proposal $q(x, \cdot)$
2. With probability
$$\alpha(x, y) = \min\left[1, \frac{\pi^*(y|d)}{\pi^*(x|d)}\frac{q(y, x)}{q(x, y)}\right],$$
accept y, then go to step 3. Otherwise reject y setting $X_{t+1} = x$, then go to step 4
3. With probability
$$\beta(x, y) = \min\left[1, \frac{\pi(y|d)}{\pi(x|d)}\frac{\pi^*(x|d)}{\pi^*(y|d)}\right],$$
accept y setting $X_{t+1} = y$. Otherwise reject y setting $X_{t+1} = x$.
4. Adaptively update the μ_B and Σ_B according to $A(X_{n+1})$ and $A^*(X_{n+1})$.
5. repeat

Convergence

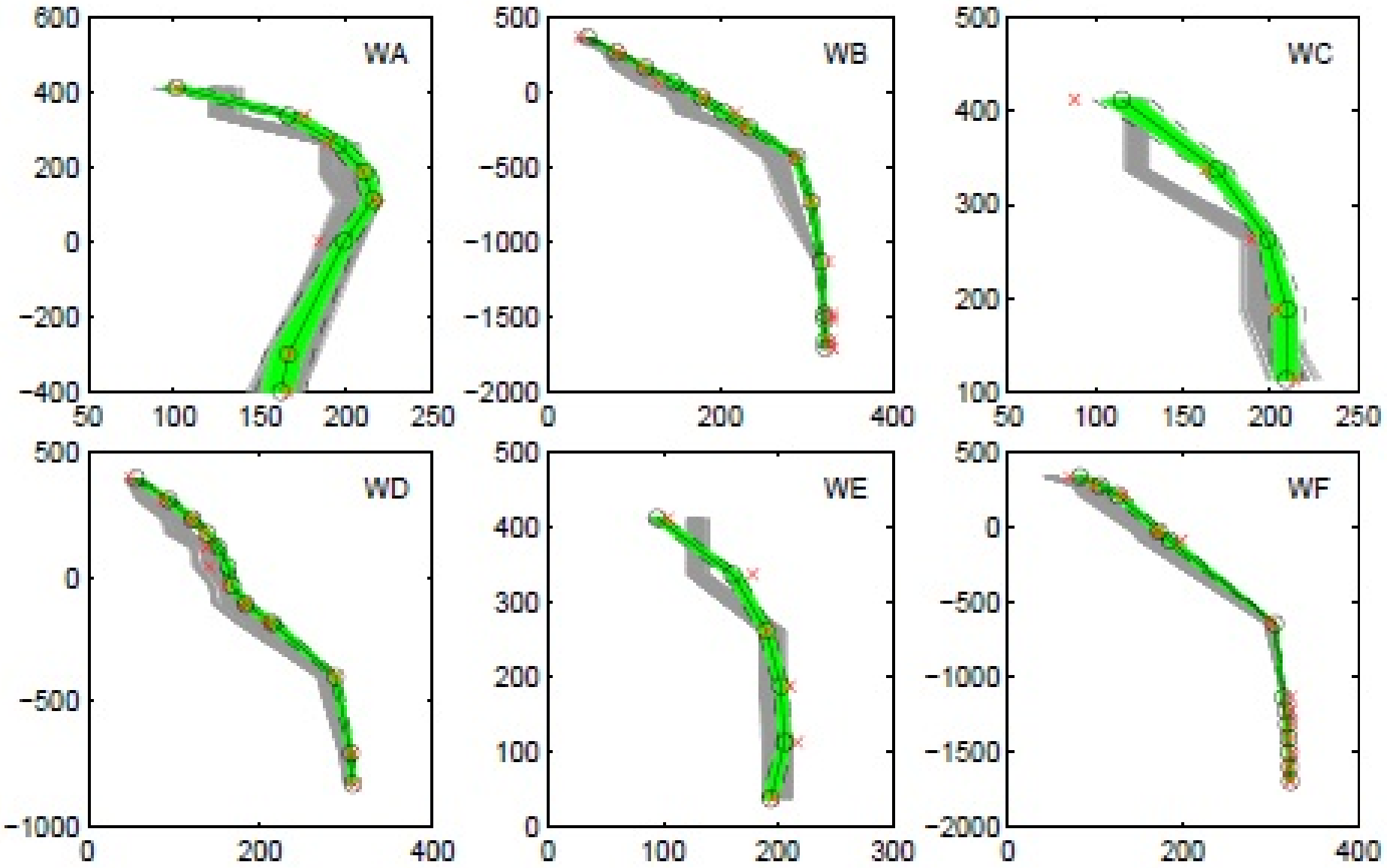
The ergodicity can be shown by applying the theorems of Roberts and Rosenthal (2007).

Results

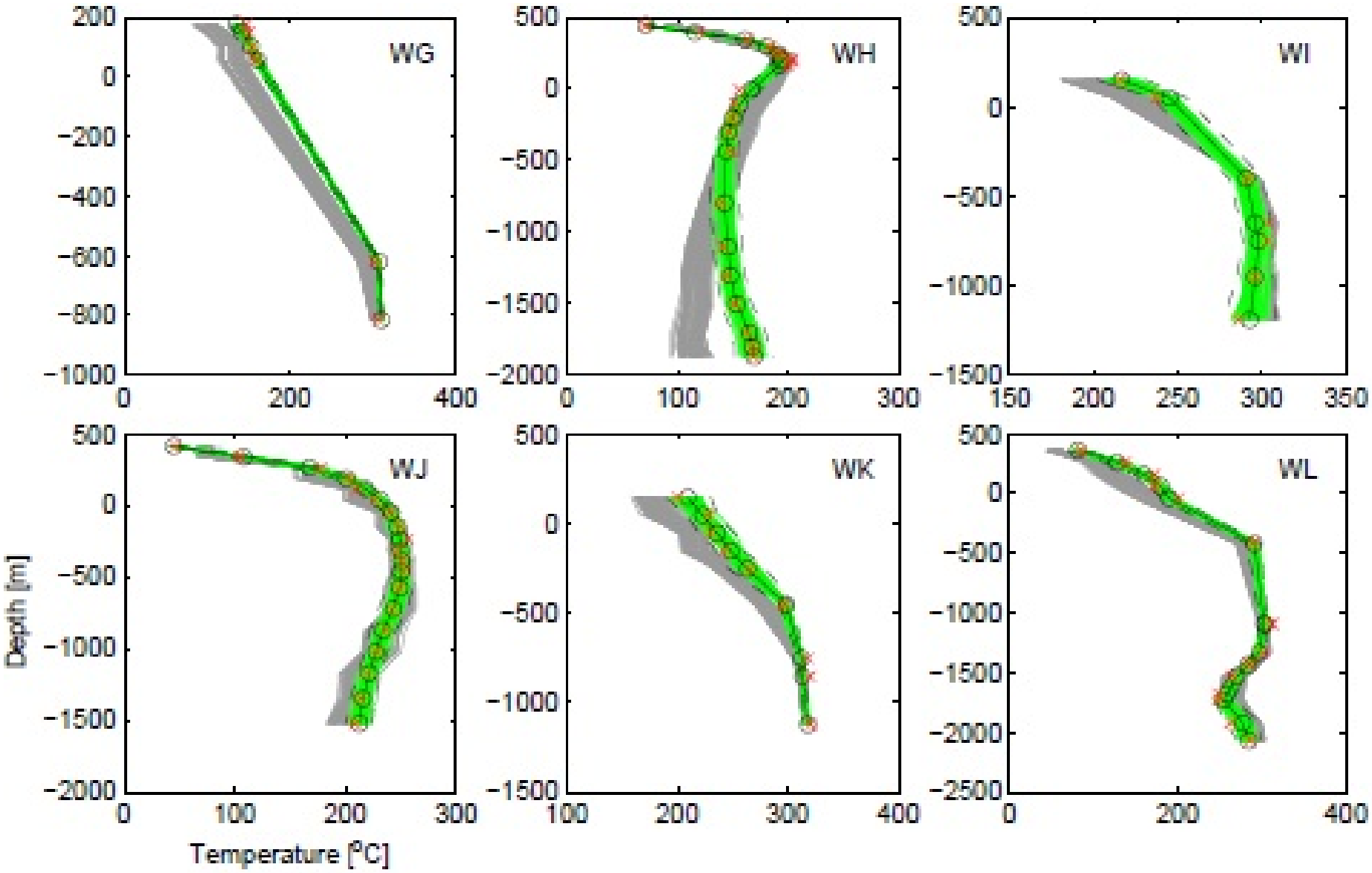


- Run ADA for 40 days, generated 23 effective samples, would cost standard MH 10 months.
- Second acceptance rate is about $0.7 - 0.8$, without error model is about 0.25 (The chain does not mix), using offline construction is about $0.4 - 0.5$. Other numerical experiments suggests similar scaling factors.

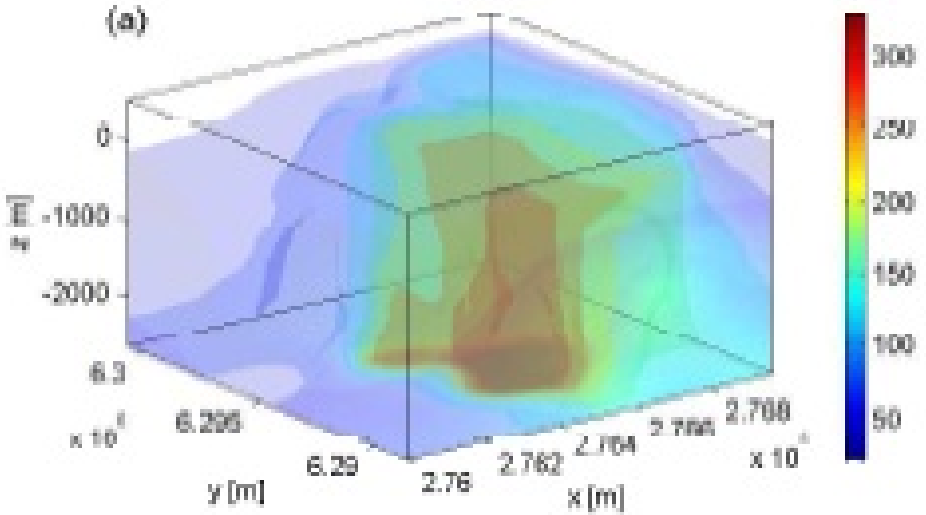
Application: Temperature Match

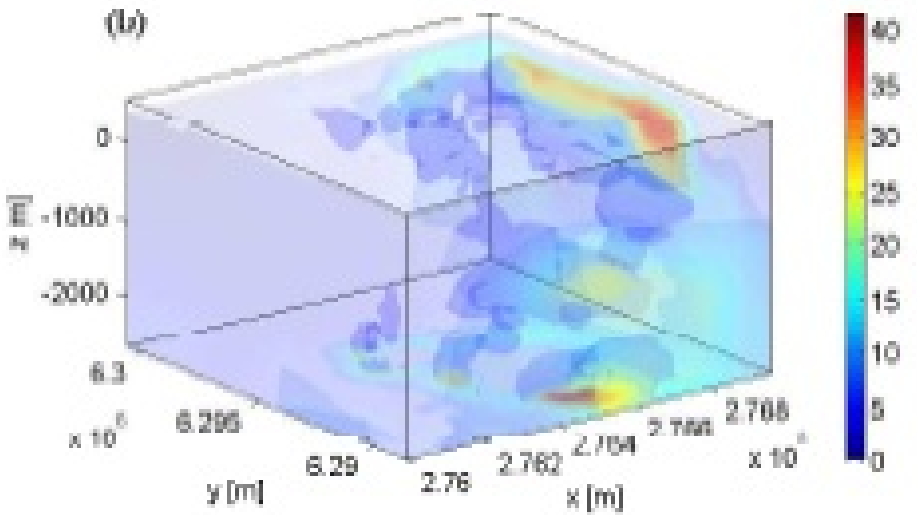


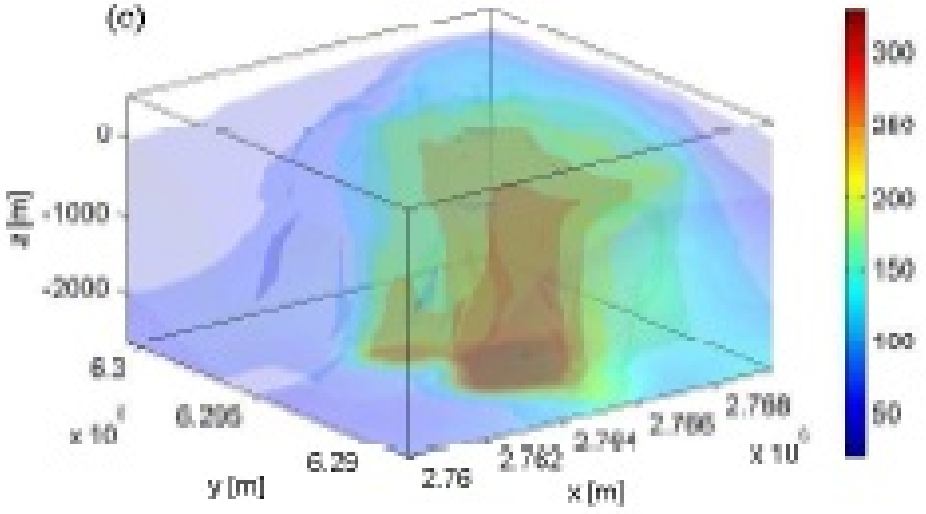
Application: Temperature Match

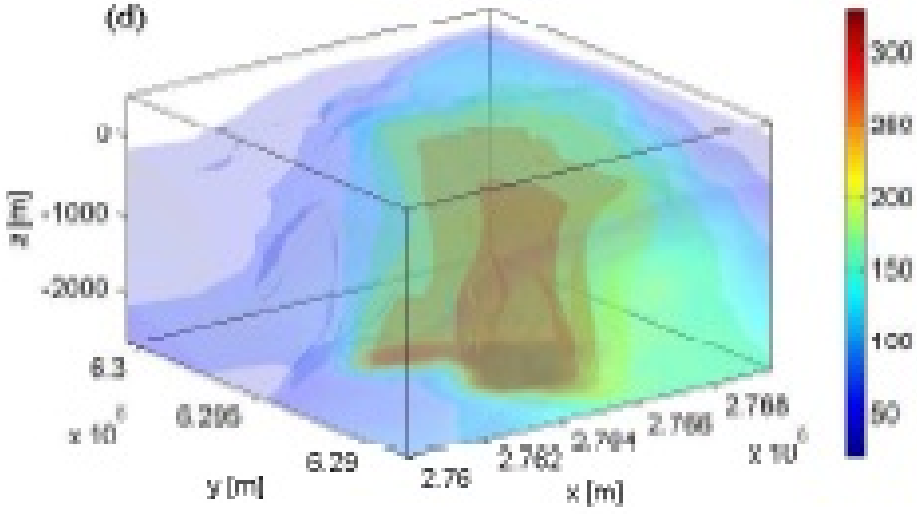


Application: Temperature distribution

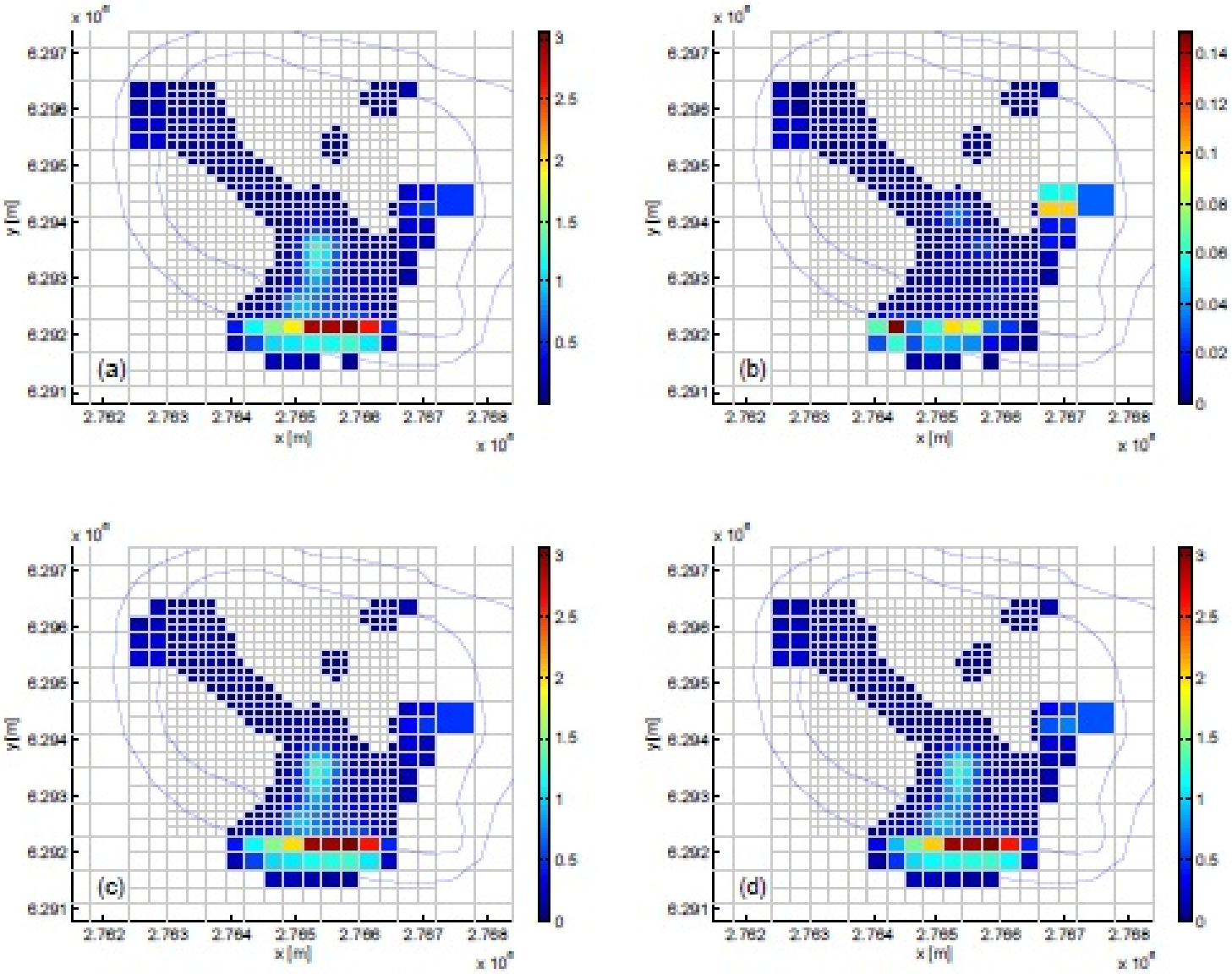




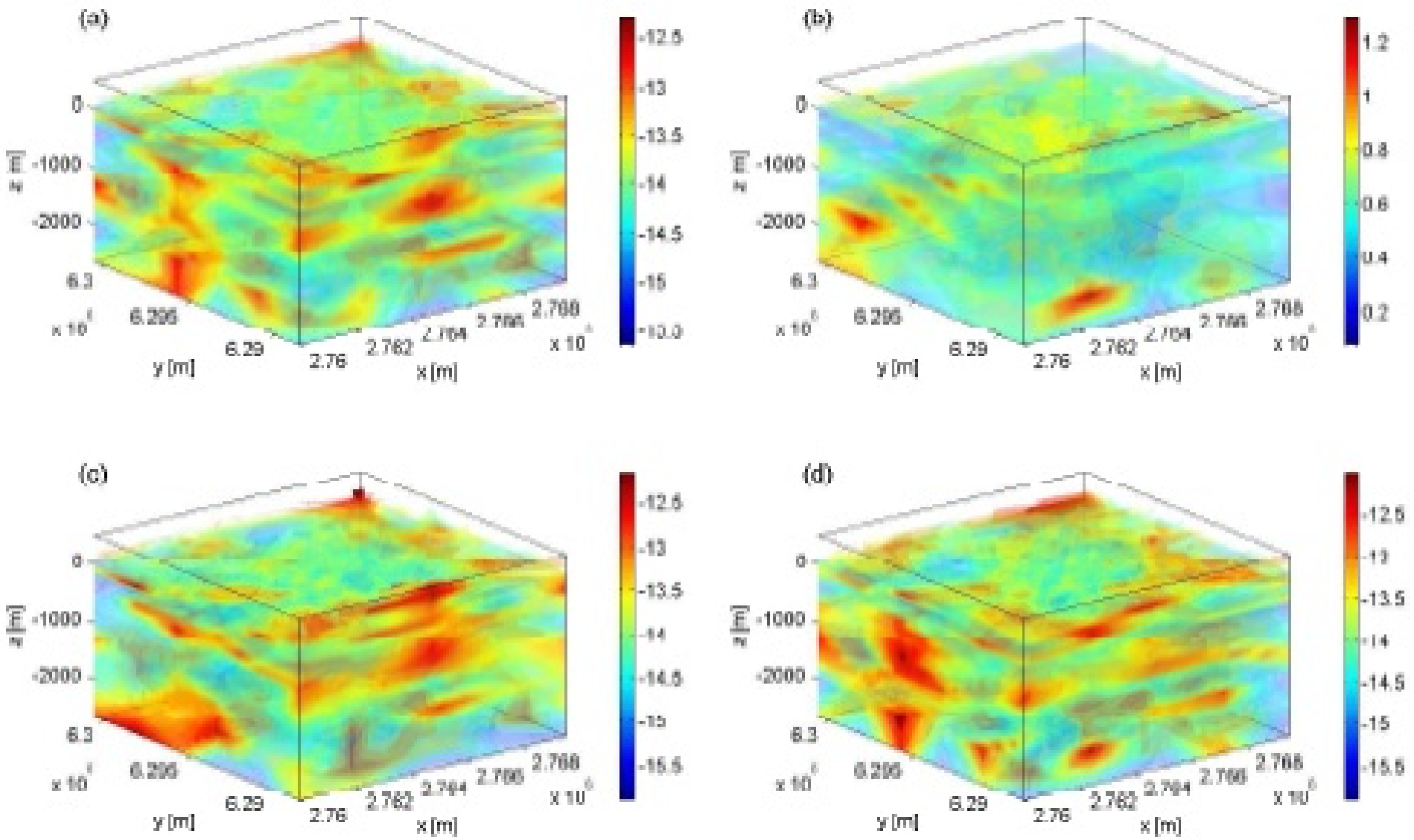




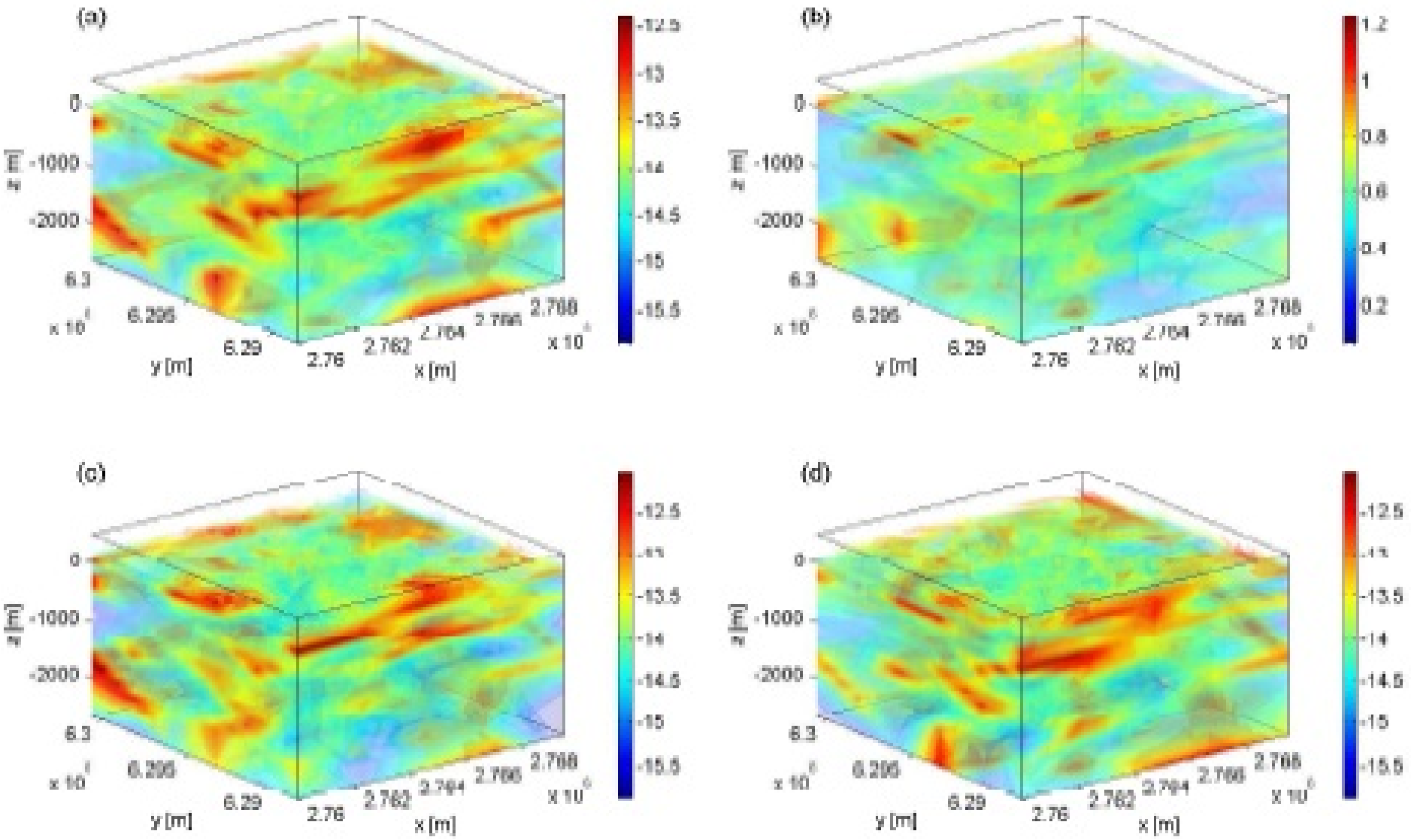
Application: Boundary Condition: Mass Input



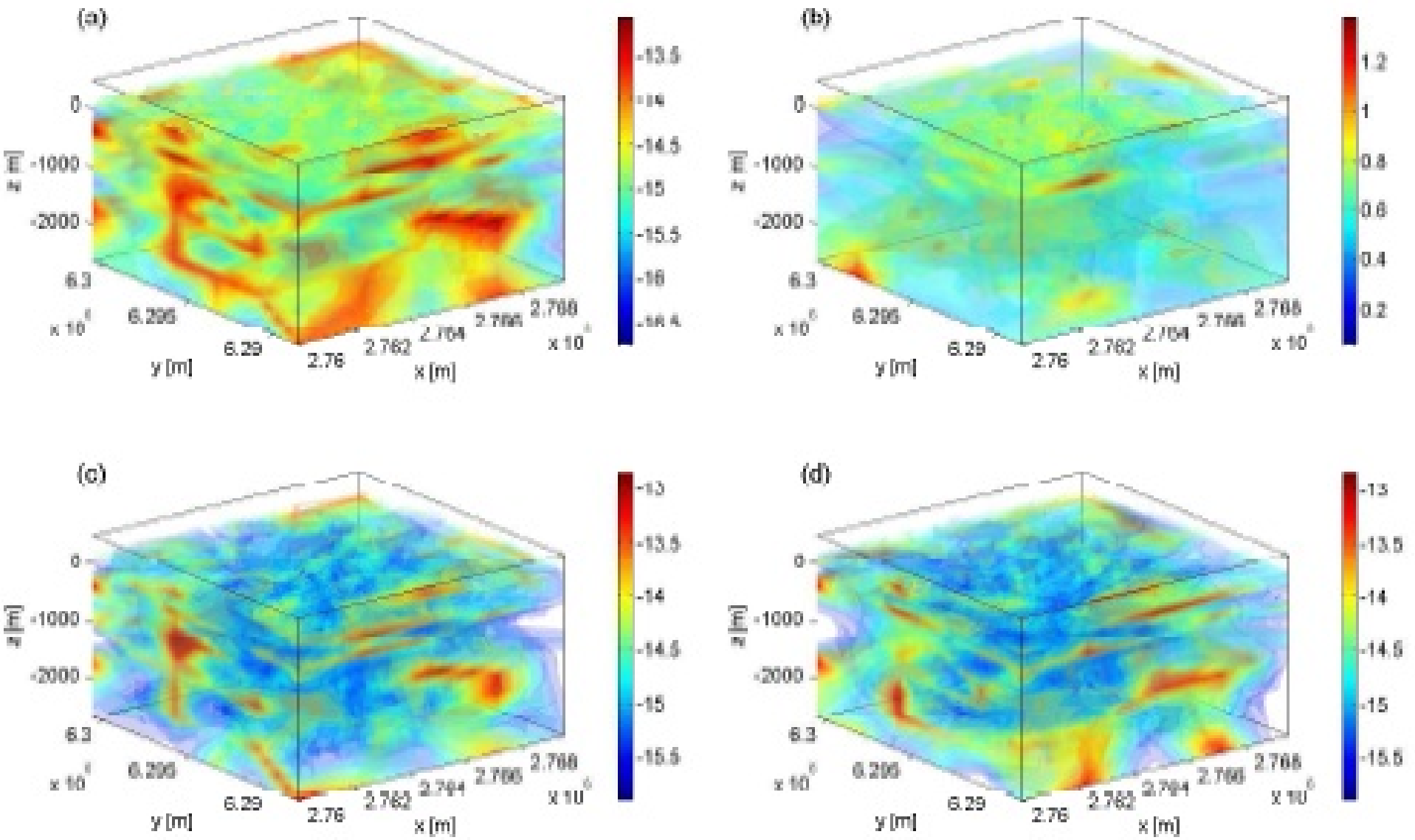
Application: Permeability: x



Application: Permeability: y



Application: Permeability: z



Outline

Motivation

Aim

1. From previous study, we know that using a fast approximation helps.
2. We want to use a reduced order model rather than coarse scale model.
3. **constructing approximation w.r.t. the posterior** is important: no offline cost, more accurate.
4. Combing the reduced basis approach with adaptive sampling.

Some Useful Facts

1. In solving an inverse problem, the model outputs have to be consistent with the data (i.e., in the support of the posterior). More informative data we have, the less variation in the outputs.
2. We would expect the model outputs lives in some low dimensional manifold.
3. MCMC sampling could track this low dimensional manifold.

Projection Based ROM

Consider the Poisson's Equation $-\nabla \cdot (x \nabla u) = f$. Discretize it, we have

The Full Model

$$\begin{aligned} A(x)u &= f \\ d &= Cu \\ \hat{d} &= d + \epsilon \mathcal{N}(0, I_{N_d}) \end{aligned} \tag{1}$$

The Reduced Order Model

$$\begin{aligned} A_r(x)u_r &= f_r \\ d_r &= C_r u_r \\ \hat{d} &= d_r + t(x) + \epsilon \mathcal{N}(0, I_{N_d}) \end{aligned} \tag{2}$$

Given Reduced Basis V for the state variable, we have $u \approx V u_r$.

- $x \in \mathcal{R}^{N_p}$
- $u \in \mathcal{R}^{DoF}$
- $d \in \mathcal{R}^{N_d}$
- $A_r(x) = V^T A(x) V$
- $f_r = V^T f$
- $C_r = CV$
- $u_r \in \mathcal{R}^{RDoF}$
- $d_r \in \mathcal{R}^{N_d}$

Methods for compute the reduced Basis includes POD (Wang and Zabaras, 2004), greedy sampling (Lieberman, Willcox and Ghattas, 2010)

Variations of POD

Offline Construction

Typical POD builds the basis w.r.t. to the prior distribution or some given physical bounds

$$E_{\pi_{pr}(x)}\left[u(x)u(x)^T\right] v_i = \lambda_i v_i$$

where E is the expectation, and v_i defines its the eigenfunction (which is a function over x, the problem domain).

Samples from the prior is problematic, especially in high dimensional cases.

Data Oriented

In our data-oriented settings, we build the basis w.r.t. to the posterior distribution

$$E_{\pi(x|d)}\left[u(x)u(x)^T\right] v_i = \lambda_i v_i$$

To control the error in the online construction, we need a good error indicator / estimator.

Error Indicator

True Error

$$t(x) = Cu(x) - CVu_r(x)$$

Requires solving the full model to get $u(x)$, we normalize the observation operator with measurement noise.

Residual

The residual

$$r(x) = f - A(x)Vu_r(x)$$

Gives a bound to the true error, without evaluating the full model, i.e.

$$\|t(x)\| \leq \|C\|\|A(x)^{-1}\|\|r(x)\|$$

- True error is not a feasible choice of error estimation, since the full model is required.
- Using residual as an error indicator has two potential issues: 1) the scaling issue, 2) it does not reflects the error of the model outputs.
- We use the approximated dual weighted residual.

Error Indicator

Dual Weighted Residual

The dual is defined as

$$\gamma(x) = A(x)^{-T} C^T$$

Then the model output can be given as $u = \vec{\gamma}^T f$. Define the true error and the residual as

$$t(x) = Cu(x) - CVu_r(x)$$
$$r(x) = f - A(x)Vu_r(x)$$

We can show that $t(x) = \gamma(x)^T r(x)$ by

$$\begin{aligned} \gamma^T r &= CA^{-1}[f - AVu_r] \\ &= Cu - CVu_r \end{aligned}$$

The dual γ provides a way to quantify the impact of residual on the true error. However, compute the exact dual is not feasible.

Error Indicator

Approximated Dual Weighted Residual

Meyer and Matthies (2003) approximate the dual by using a ROM that has higher order of accuracy. In our setting, since we assume all the model outputs are similar to each other, the dual defined at the MAP point would provide a resealable approximation

$$\hat{\gamma} = \gamma(x_{MAP})$$

and hence the approximated true error $\hat{t}(x) = \hat{\gamma} r(x)$

We can also use the ensemble of full model evaluations to build a library of dual. Then possibly to provide a error estimation by

- point estimate from the closest full model evaluation rather than MAP.
- or an average dual from all the full model evaluations.

All these error indicator quantifies that given a new (MCMC) proposal, if the ROM evaluated at the proposal is accurate enough, up to some threshold.

Exact Algorithm

Suppose the chain is at $X_n = x$, given a proposal $q(x, \cdot)$, a ROM and some number N_{lag}

1. Simulate a sub-MCMC chain (denoted by Z_i) that with the approximate posterior defined by the ROM, until either the sub-chain exceed the iteration count N_{lag} or the error $\|\hat{t}\|_\infty > \epsilon$ for the latest accepted state Z_i.
2. Use the last state Z_i as a proposal in the delayed acceptance, evaluate the full model at Z_i, and the acceptance probability $\beta(x, Z_i)$. Given an random number τ,
   - If $\tau < \beta$, Set $X_{n+1} = Z_i$, and update the ROM if $\|\hat{t}(Z_i)\| > \epsilon$
   - otherwise Set $X_{n+1} = x$
3. Some adaptation on the proposal and N_{lag}.

Remarks

1. This algorithm is exact, but need many forward model evaluations, i.e., it is unbiased, but the variance reduction is not very good in terms of number of FOM evaluations.

Approximate Algorithm

Suppose the chain is at $X_n = x$

1. propose y from $q(x, \cdot)$. Evaluate the ROM, get reduced state $u_r(y)$ and error $\hat{t}(y)$
2. 1. If $\|\hat{t}(y)\|_\infty \geq 1$, evaluate the full mode to get $u(y)$, then get the acceptance probability $\alpha(x, y)$. Accept/reject according to α.
   2. If $\|\hat{t}(y)\|_\infty \in (\epsilon, 1)$, run one step of the delayed acceptance using $u_r(y)$ as an approximation
      - If the approximation is accepted, Evaluate the full model to get $u(y)$, and hence $\beta(x, y)$. Accept/reject according to β.
      - Otherwise reject.
   3. If $\|\hat{t}(y)\|_\infty \leq \epsilon$, we treat the ROM as good approximation to, then $\alpha(x, y)$ is approximated. Accept/reject according to α.

For every Full model evaluation, we update the ROM.

1. If the error is greater than 1 (the standard deviation of the noise), use the full model
2. If the error is within the threshold ϵ and 1, it is considered as an "reasonable" approximation, then the delayed acceptance scheme is used to make the correction.
3. For error $< \epsilon$, the ROM is considered to be the same as the full model. This gives the main computational gain.

Approximate Algorithm

Mean Square Error

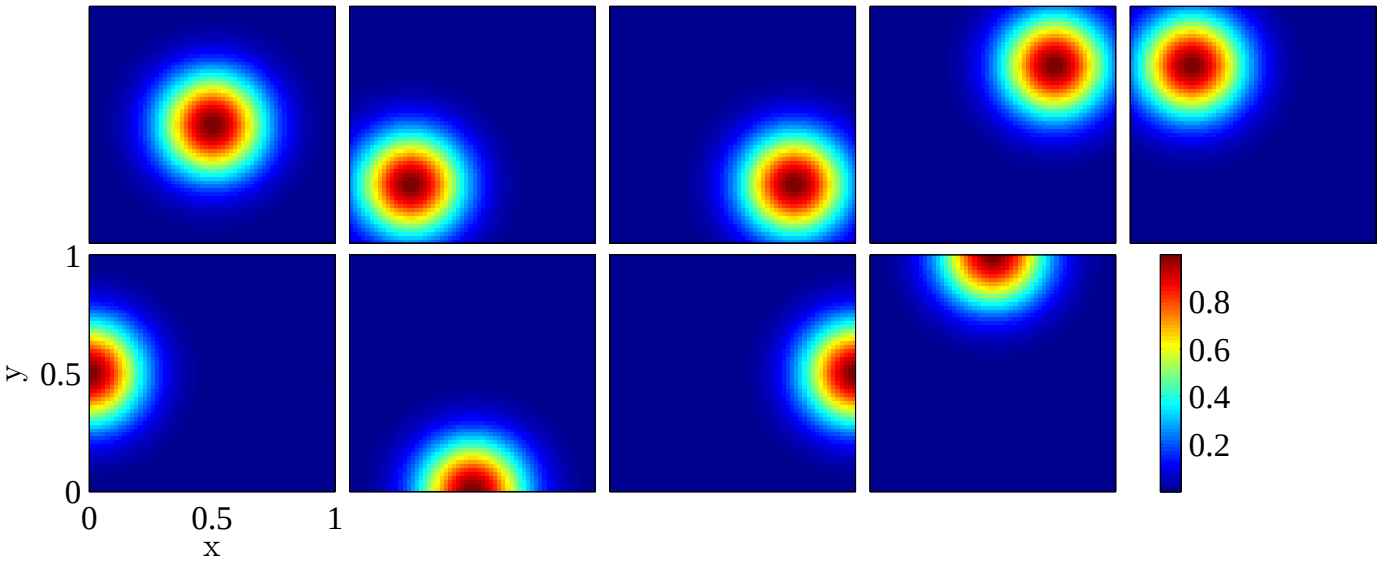
Recall that the mean square error of some estimator $\hat{\theta}$ to the unknown quantity of interest θ

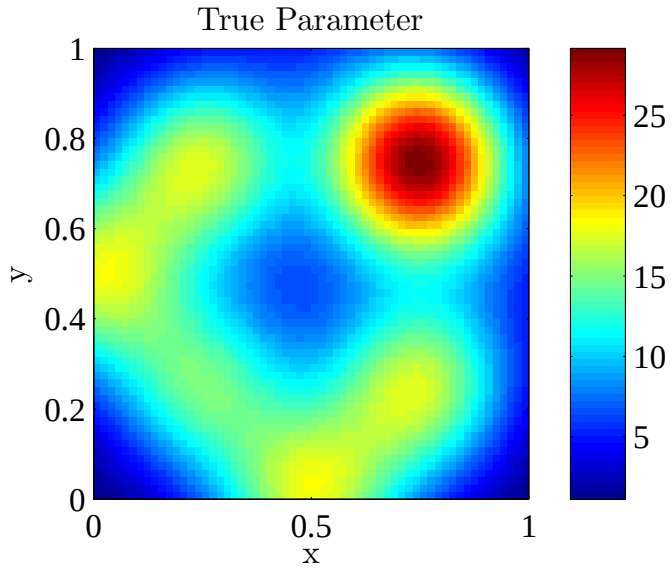
$$MSE(\hat{\theta}) = Var(\hat{\theta}) + (E[\hat{\theta} - \theta])^2 = Var(\hat{\theta}) + Bias(\hat{\theta})^2$$

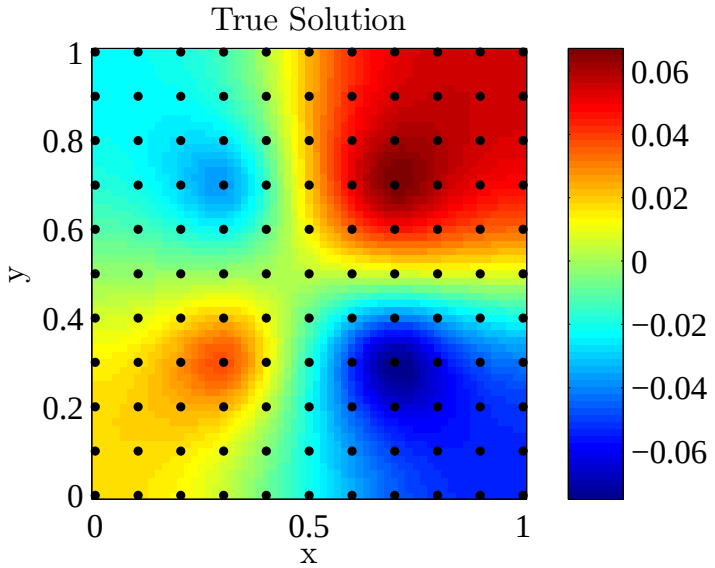
1. Approximated MCMC provided a biased estimator that has better variance reduction (as we can compute more samples in the same CPU time),
2. The exact MCMC (using delayed acceptance) provides an unbiased estimator that has larger variance.
3. The bias of the previous algorithm is still tractable, as the error is controlled by ϵ.
4. We test various choice of ϵ, for $\epsilon = 10^{-3}$, the approximate algorithm is very accurate. See the coupling test.

A 9D Test Case: Problem Setup

9 parameters are defined by basis functions. 100 measurements with signal to noise ratio 50.







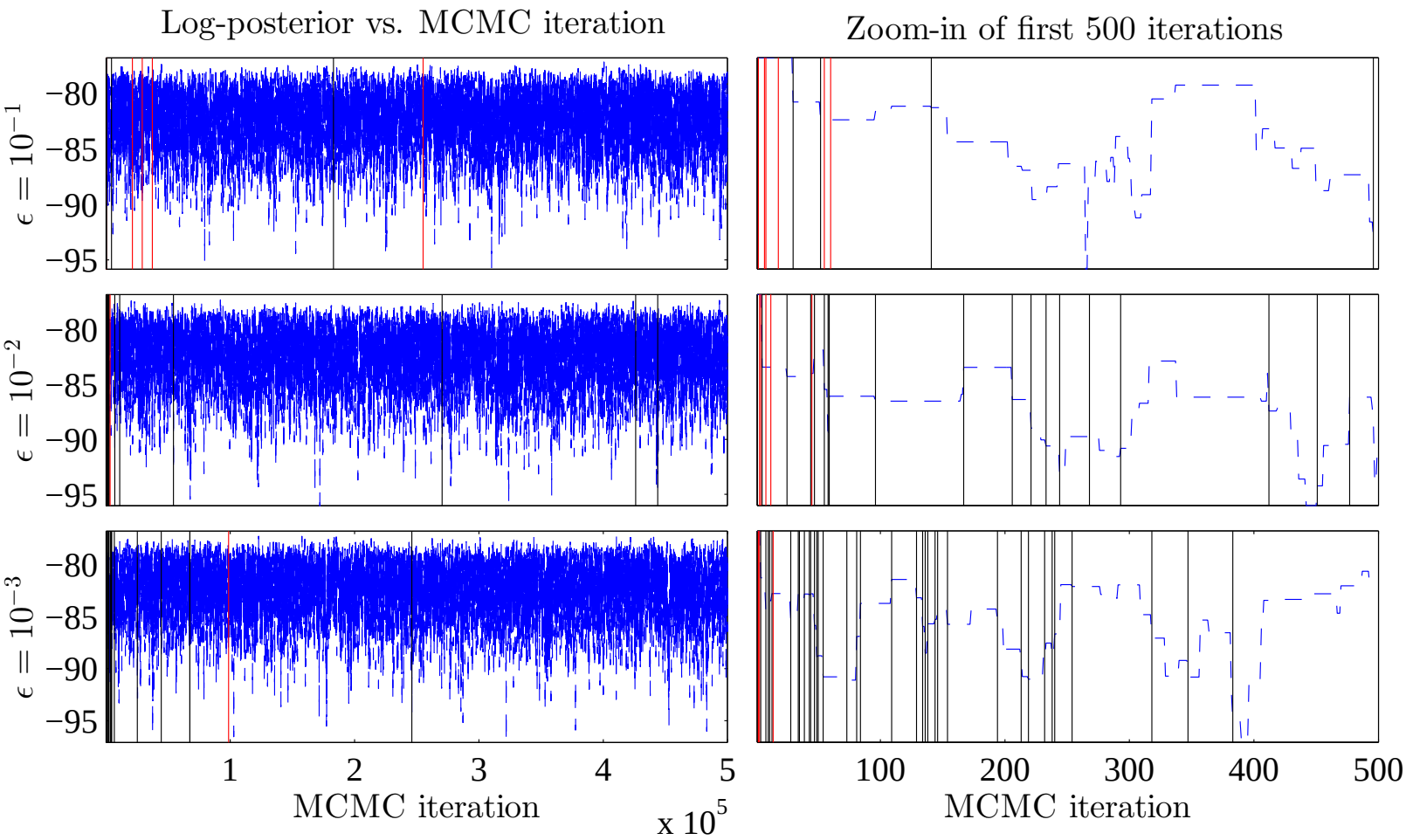
A 9D Test Case: Numerical Results

Comparison of Efficiency

	Exact			Approximate			1 Step MH
MCMC iterations	10^4, $N_{lag} = 50$			5×10^5			5×10^5
ϵ	10^{-1}	10^{-2}	10^{-3}	10^{-1}	10^{-2}	10^{-3}	-
Average β	0.9138	0.9562	0.9653	-	-	-	-
FOM evaluations	10^4	10^4	10^4	17	37	59	5×10^5
ROM basis	13	34	61	17	37	59	-
CPU time (sec)	329.3	370.7	425.2	119.5	157.5	195.5	1.032×10^4
ESS	2195	2426	2557	2443	2372	2412	2016
ESS / CPU time	6.666	6.544	6.159	20.44	15.06	12.34	0.1953
Speed-up factor	34.13	33.51	31.54	104.7	77.11	63.18	1

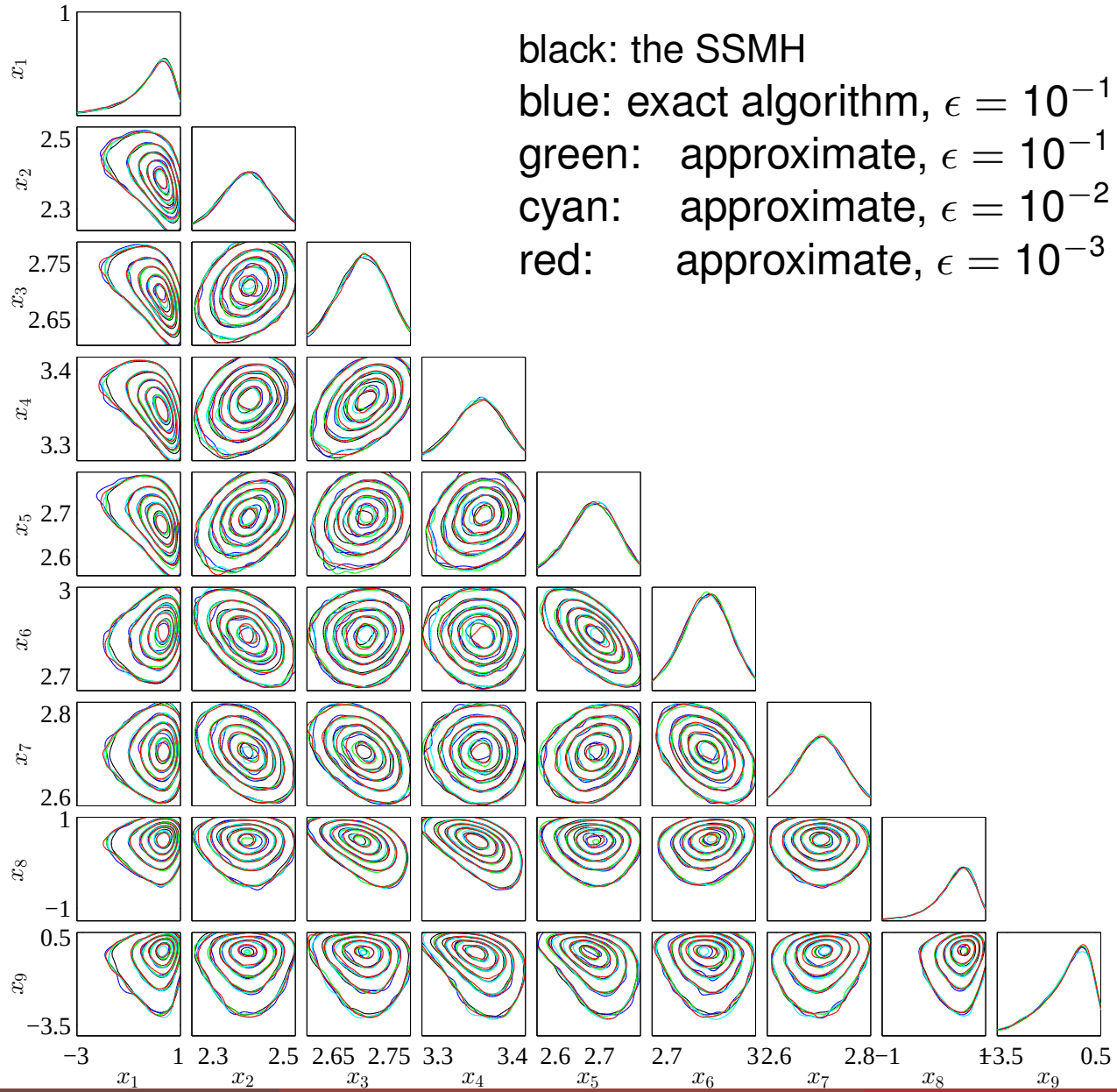
Comparison of the computational efficiency of the exact algorithm, the approximate algorithm, and the single step MH (SSMH) algorithm. For the exact algorithm and the approximate algorithm, three examples with different values of $\epsilon = 10^{-1}, 10^{-2}, 10^{-3}$ are given. Fixed proposal distribution is used in all the test cases.

A 9D Test Case: Numerical Results

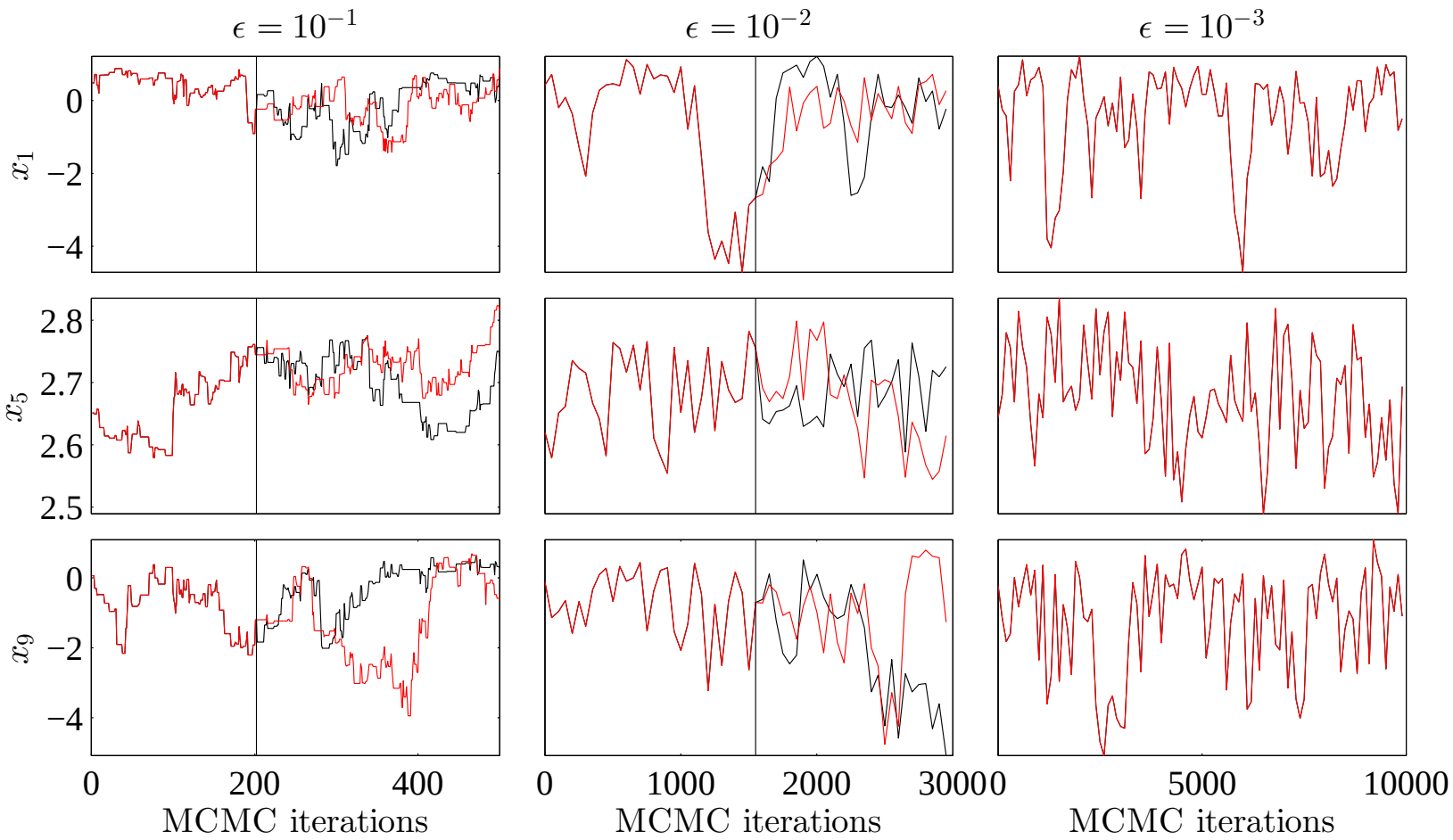


The trace of the log-posterior against MCMC iterations. From top to bottom: $\epsilon = 10^{-1}, 10^{-2}, 10^{-3}$. The red and black lines indicate FOM evaluations, where red means a rejected proposal, and black means an accepted proposal.

A 9D Test Case: Numerical Results

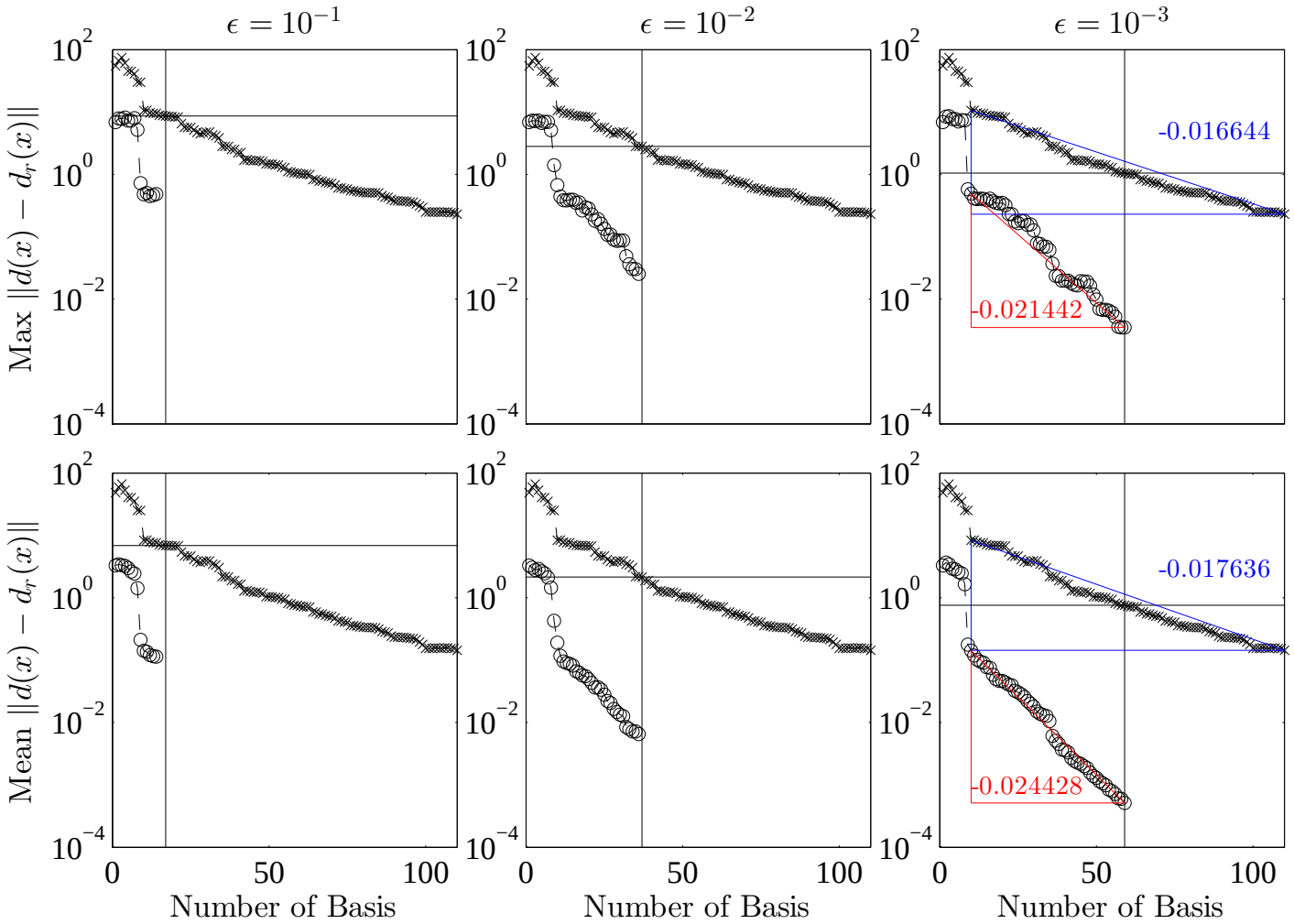


A 9D Test Case: Coupling Time



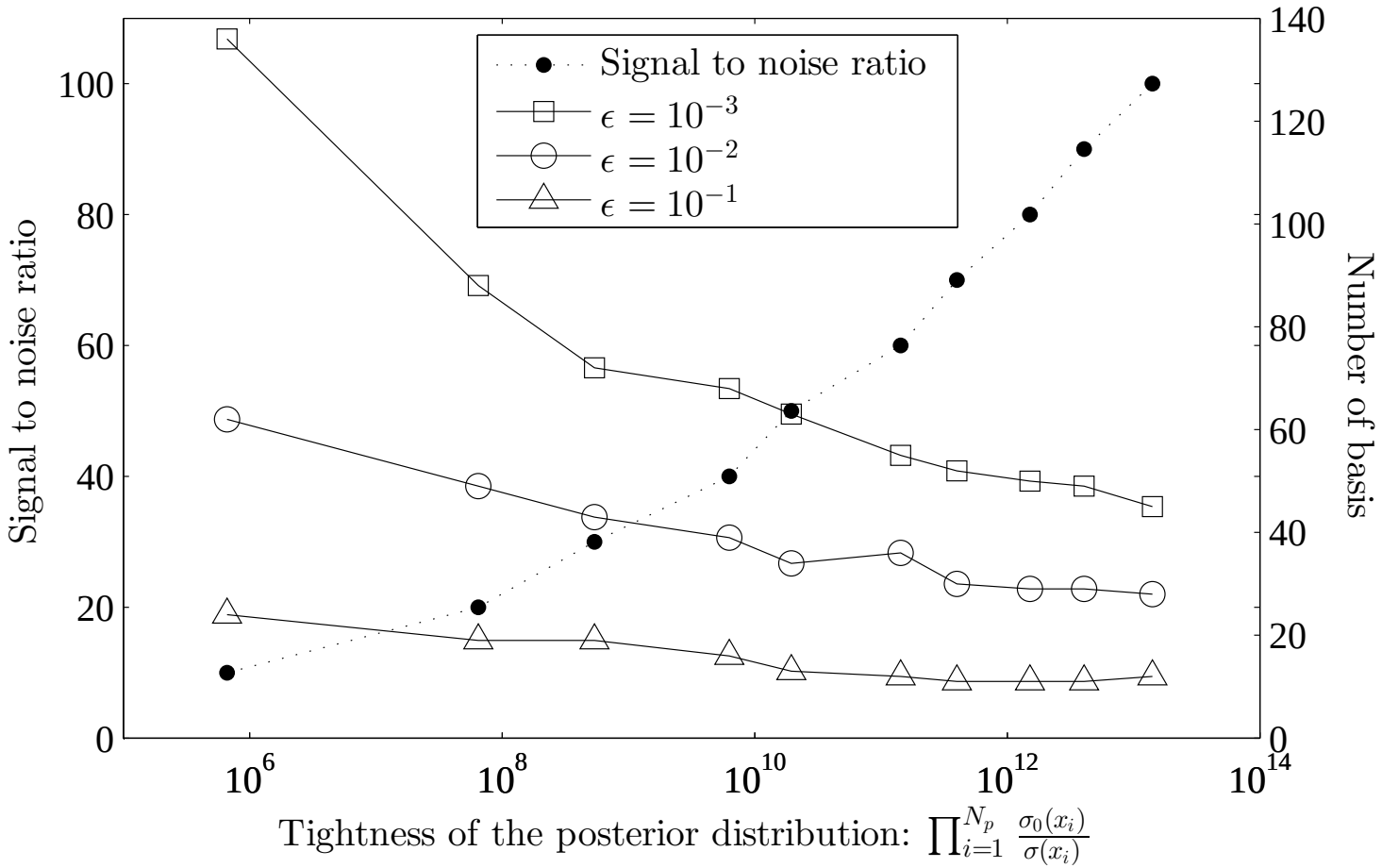
Coupling time between the SSMH algorithms sampling the approximate posterior and the SSMH sampling the exact posterior. From left to right, the approximate posterior uses ROM that constructed with different error threshold, $\epsilon = 10^{-1}, 10^{-2}, 10^{-3}$.

A 9D Test Case: Error in the ROM



Comparison of the true error of the data-oriented ROM with the ROM built w.r.t. the prior. From left to right: data-oriented ROMs with threshold, $\epsilon = 10^{-1}, 10^{-2}, 10^{-3}$.

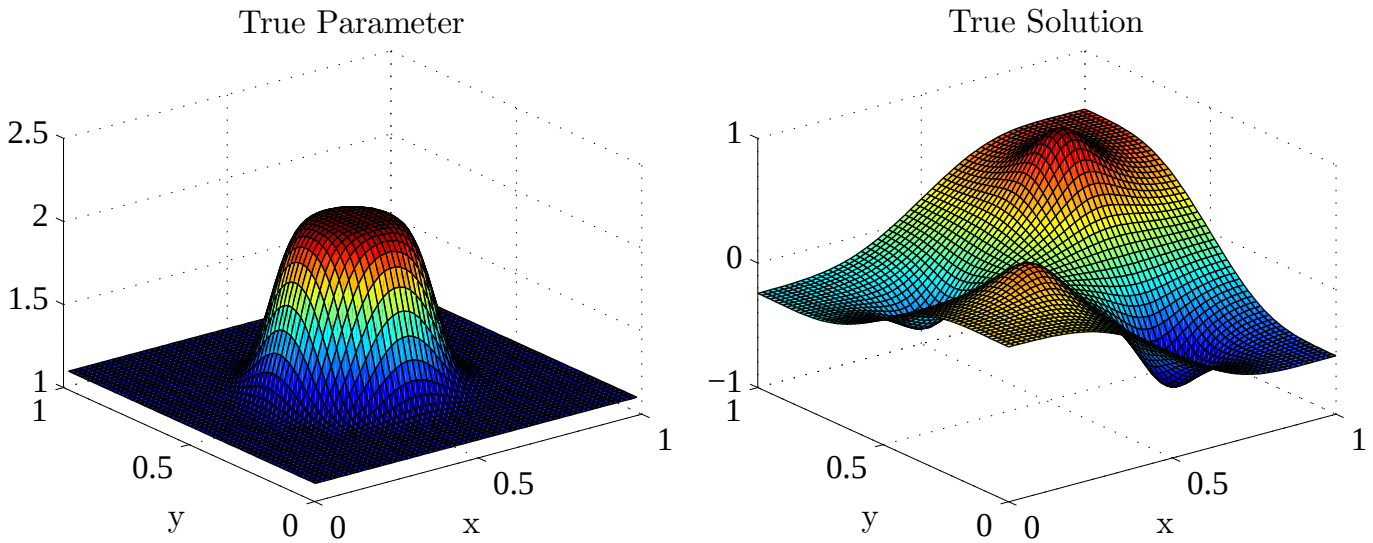
A 9D Test Case: Influence of the Likelihood



By changing the signal to noise ratio, we can control the amount of information carried in the data. We test the number of reduced basis versus the amount of information in the data.

A High Dimensional Example: Problem Setup

Distributed parameter, prior defined by the sparse precision matrix.
KL expansion is used to truncate the prior at 99.99% energy, gives 53 basis.



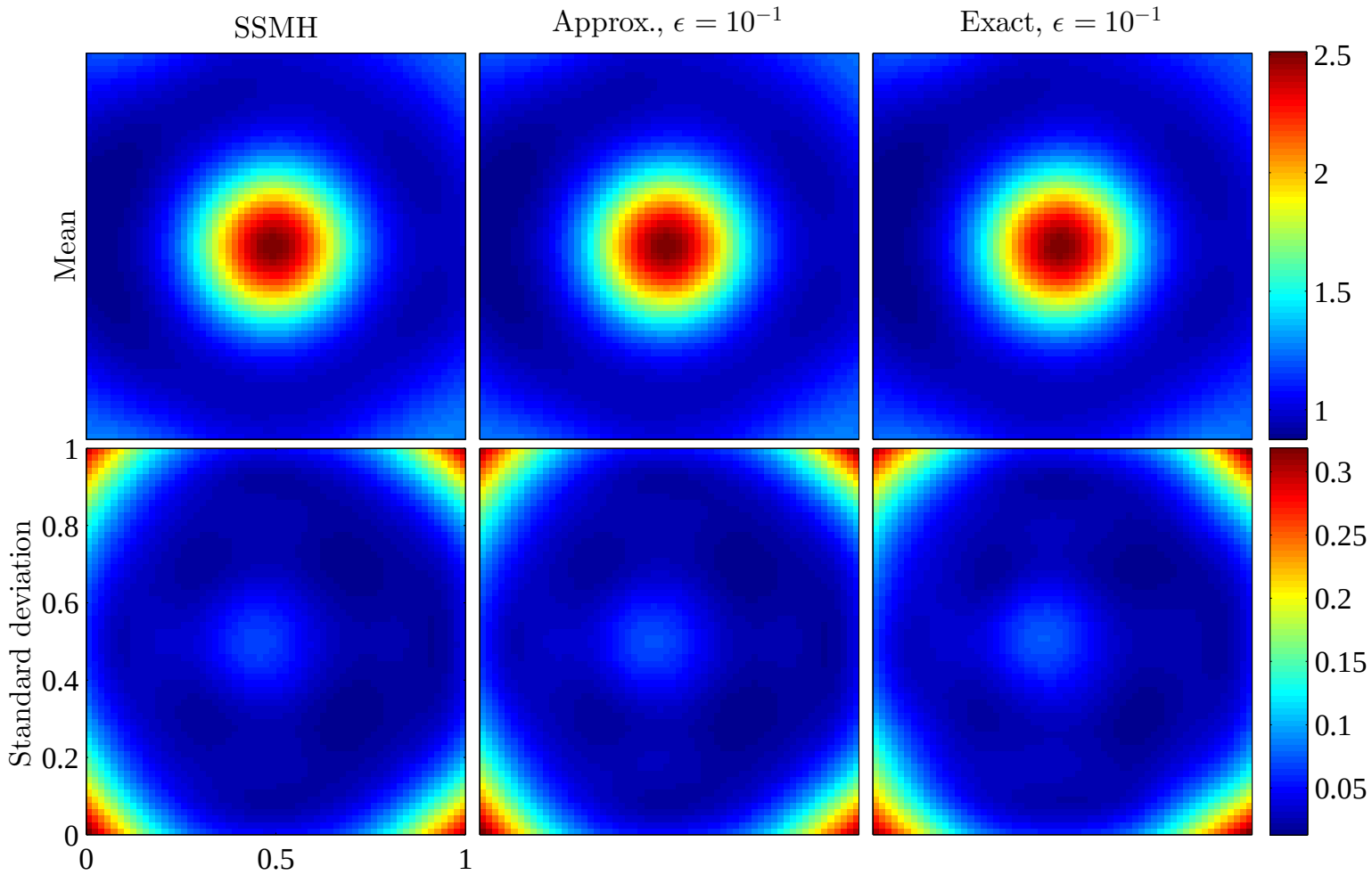
A High Dimensional Example: Numerical Results

Comparison of Efficiency

	Exact	Approximate			1 Step MH
MCMC iterations	10^4, $N_{lag} = 50$	5×10^5			5×10^5
ϵ	10^{-1}	10^{-1}	10^{-2}	10^{-3}	-
Average β	0.9271	-	-	-	-
FOM evaluations	10^4	61	123	198	5×10^5
ROM basis	62	61	123	198	-
CPU time (sec)	673.2	483.0	1023	1849	2.266×10^4
ESS	2221	2468	2410	2445	2472
ESS / CPU time	3.299	5.110	2.356	1.322	0.1091
Speed-up factor	30.23	46.84	21.59	12.12	1

Comparison of the computational efficiency of the exact algorithm, the approximate algorithm, and the single step MH (SSMH) algorithm. For the exact algorithm, $\epsilon = 10^{-1}$ is used. For the approximate algorithm, three different values of $\epsilon = 10^{-1}, 10^{-2}, 10^{-3}$ are used. We use a fixed proposal distribution in all the test cases.

A High Dimensional Example: Numerical Results



Mean and standard deviation at each spatial location of the permeability field.

Thanks

Discussion

- Using adaptation is important, but this has not been fully discovered.
- POD does not reflect the multiscale behavior of the physics, different forms of reduced order models?
- Nonlinear PDEs using discrete empirical interpolation (DEIM)
- Applications in dynamic systems
- Extension to using high order derivatives.

Thanks!

A Real Volcano