Adaptive Multilevel Delayed Acceptance

Computational Uncertainty Quantification: Mathematical Foundations, Methodology Data (Thematic Programme)

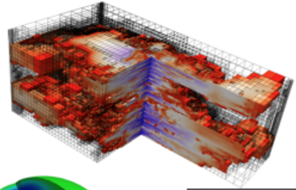
Prof. Tim Dodwell (Exeter/Turing/digiLab)

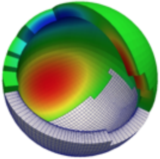
Spring 2022

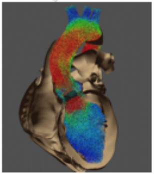
work funded by Turing AI Fellowship

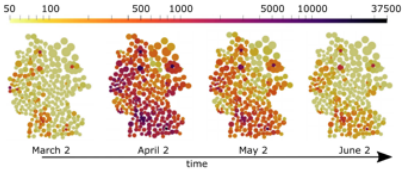
From Models to Decisions . . .

- Huge explosion of ‘data-driven’ methods!
- Huge explosion of High Performance Simulations!
- What do ‘industry’ really want / care about?
    - The perfect model? → **No!**
    - A high dimensional output → **Often Not!**
    - Understanding of what happens on average → **Often not!**
- I want models and data to sing together!
- I want predictions to revert to our scientific knowledge of physics in the absense of data - with an appropriate level uncertainty.









Adaptive Multilevel Delayed Acceptance - a team sport!

- Mikkel Lykeggaard (Exeter)
- Colin Fox (Otago)
- Rob Scheichl (Heidelberg)

Bayesian Inverse Problems

- We have (limited) observations of a system

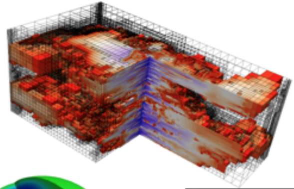
$$\mathbf{d} \in \mathbb{R}^M$$

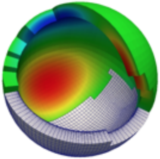
- A (mathematical) model $\mathcal{F}(\theta) : \mathbb{R}^Z \to \mathbb{R}^M$ which predicts our data given parameters θ.
- We connect our model and data

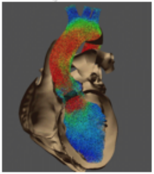
$$\epsilon = \mathbf{d} - \mathcal{F}(\theta) \sim \mathcal{N}(\mathbf{0}, \sigma_\epsilon^2 \mathbb{I})$$

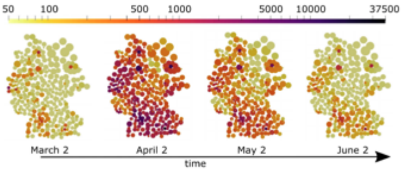
- We have some *prior* of parameters - $\pi(\theta)$.
- We wish to find the distribution of parameters given our observations - $\pi(\theta|\mathbf{d})$
- Quantity of Interest is functional $Q(\theta)$ which to compute statistics, e.g.

$$\mathbb{E}_{\pi(\theta|\mathbf{d})}[Q(\theta)]$$









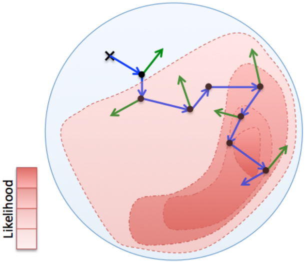
Markov Chain Monte Carlo - Metropolis-Hastings

Algorithm 1. Metropolis-Hastings (MH)

- Given $\theta^{(j)}$, generate a proposal ψ distributed as $q(\psi|\theta^{(j)})$,
- Accept proposal ψ as the next state, i.e. set $\theta^{(j+1)} = \psi$, with probability

$$\alpha(\psi|\theta^{(j)}) = \min\left\{1, \frac{\pi_t(\psi)q(\theta^{(j)}|\psi)}{\pi_t(\theta^{(j)})q(\psi|\theta^{(j)})}\right\} \tag{1}$$

otherwise reject ψ and set $\theta^{(j+1)} = \theta^{(j)}$.



Markov Chain Monte Carlo - Metropolis-Hastings

The **Good Things** **Metropolis-Hastings**

- **Simple!**
- Alg. 1 simulates a fixed (stationary) transition kernel $K(y|x)$
- Repeated iterations generate a (homogeneous) Markov chain.
- **MH** (Alg. 1) is in detailed balance with π_{t}, i.e.

$$\pi_{\mathrm{t}}(x) K(y|x) = \pi_{\mathrm{t}}(y) K(x|y),$$

- Mild conditions on $q(\cdot|\cdot)$ and start point, $\Theta := \{\theta^0, \theta^1, \ldots, \theta^N\} \sim \pi_t$

The **Big Challenges** with **Metropolis-Hastings**

1. Evaluating π_t - can be **computationally expensive**!
2. **Markov Chain** is strongly correlated $\Theta := \{\theta^0, \theta^1, \ldots, \theta^N\}$.
3. **Difficult to Parallelise** - fundamental challenge since by their nature Markov processes are **sequential**.

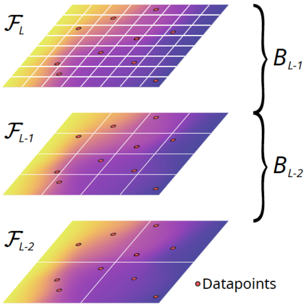
Exploiting Hierarchies of Models

Lemma 1. If the proposal transition kernel $q(\cdot|\cdot)$ in Alg. 1 is in detailed balance with some distribution π_C, then the acceptance probability (1) may be written

$$\alpha(\psi|\theta^{(j)}) = \min\left\{1, \frac{\pi_t(\psi)\pi_C(\theta^{(j)})}{\pi_t(\theta^{(j)})\pi_C(\psi)}\right\} \tag{2}$$

Proof Sub. the detailed balance statement $\pi_C(\psi)q(\theta^{(j)}|\psi) = \pi_C(\theta^{(j)})q(\psi|\theta^{(j)})$ into (1) to get (2), almost everywhere.

- Idea is to **exploit a hierarchy of approximate models $\mathcal{F}_\ell$**
  - Grid resolution (norm for us) / Parameters θ_ℓ / Data $\mathbf{d}_\ell$.
- Consider just **two levels** and no level dependence on θ or $\mathbf{d}$.
- Therefore have
  - Fine / Target $\pi_F \equiv \pi_t$
  - Coarse / Approximate π_C



Multilevel Markov Chain Monte Carlo - Bottom Up Approach

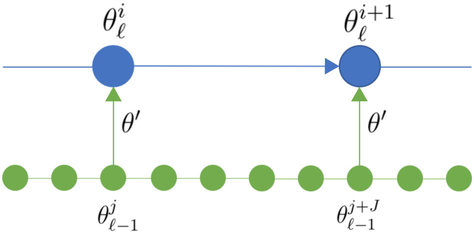
Dodwell, Ketelsen, Scheichl, and Teckentrup, Multilevel Markov Chain Monte Carlo, SIAM Rev., 61:509-545, 2019.

Two key **motivating points**

1. Use subchains generated π_C to cheaply build 'good' proposals.
2. Exploit multilevel **variance reduction mechanism** - Giles (Oxford)

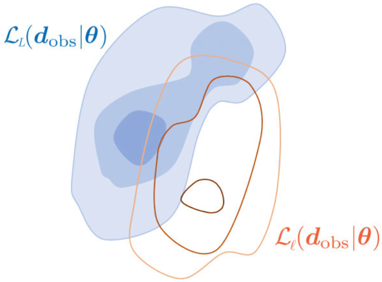
$$\mathbb{E}_{\pi_F}(Q_F) = \mathbb{E}_{\pi_C}(Q_C) + \underbrace{[\mathbb{E}_{\pi_F}(Q_F) - \mathbb{E}_{\pi_C}(Q_C)]}_{\text{Make correlated!}}$$

Algorithm in a picture



Multilevel Markov Chain Monte Carlo - The problem

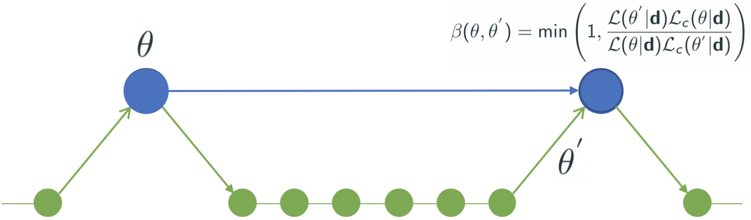
- MLMCMC is not a Markov Process!
  - If we **reject** on the fine, **coarse is not reset**
  - Theoretically only works if subsampling rate $J = \infty$
  - Works well in practise if $J > \tau$ (autocorr. length of subchain).
- **Struggles** if difference between π_F and π_C is **big**.



Adaptive Multilevel Delayed Acceptance (this work) addresses these problems!

Multilevel Delayed Acceptance - Top Down

- Run finite length subchain of random length $J \sim p(\cdot)$ (why??) on approximate level.



Coarse subchain $\Theta_c = \{\hat{\theta}_0, \hat{\theta}_2, \ldots, \hat{\theta}_J\}$ generates proposal θ'

- **Idea** cheaply generate (more) independent proposal from approximate posterior $\pi_C \sim \pi_F$
- Cost saving is approx. different in cost between $\mathcal{F}$ and $\mathcal{F}_C$ times **acceptance rate** (typically high).
- Generates a Markov Chain and can prove **detailed balance**.

Multilevel Delayed Acceptance - Step 1

Alg. 2. Randomised-Length-Subchain Surrogate Transition (RST)

Input: Fine density $\pi_{\mathsf{F}}(\cdot)$, Coarse density $\pi_{\mathsf{C}}(\cdot)$, proposal kernel $q(\cdot|\cdot)$, probability mass function over subchain length $p(\cdot)$, start state θ^0

- Draw the subchain length $n \sim p(\cdot)$.
- Starting at $\theta^{(j)}$, generate a subchain of length n using the Metropolis–Hastings Alg. 1 targeting the coarse target

$$\psi = \mathbf{MH}\left(\pi_{\mathsf{C}}(\cdot), q(\cdot|\cdot), \theta^{(j)}, n\right) \tag{3}$$

- Accept the proposal ψ as the next sample, i.e. set $\theta^{(j+1)} = \psi$, with probability

$$\alpha(\psi|\theta^{(j)}) = \min\left\{1, \frac{\pi_{\mathsf{F}}(\psi)\pi_{\mathsf{C}}(\theta^{(j)})}{\pi_{\mathsf{F}}(\theta^{(j)})\pi_{\mathsf{C}}(\psi)}\right\}. \tag{4}$$

otherwise reject and set $\theta^{(j+1)} = \theta^{(j)}$.

Multilevel Delayed Acceptance - Detailed Balance

Lemma 2 If transition kernels $K_1(x|y)$ and $K_2(x|y)$ are each in detailed balance with a distribution π, and K_1 and K_2 commute, then the composition of the kernels ($K_1 \circ K_2$) is in detailed balance with π.

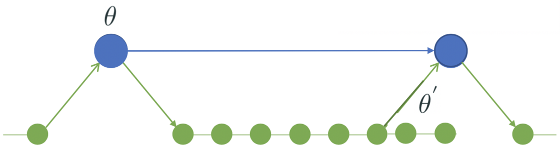
Lemma 3 Alg. 2 simulates a Markov chain that is in detailed balance with $\pi_{\mathsf{F}}(\cdot)$.

- q_C computes with itself
- By induction q_C^n (application n times) is in detailed balance with $\pi_C(\cdot)$.
- Random subchain length gives an effective mixture kernel

$$\sum_{n \in \mathbb{Z}^+} p(n) q_C^n(\cdot|\cdot)$$

- Apply **Lemma 1** $\rightarrow$ in detailed balance with π_F.

Multilevel Delayed Acceptance - Variance Reduction



- Coarse subchain $\not\sim \pi_C$ - Like mini burn ins from π_F
- Samples from "hybrid" mixture distributions

$$\tilde{\pi}_C = \frac{1}{J}\sum_{j=1}^{J} K_C^j \, \pi_F \tag{5}$$

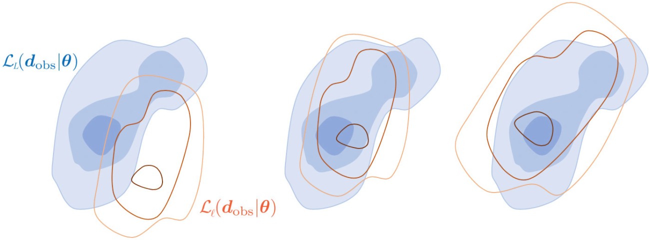
where J is max subchain length and $K_C^j = \underbrace{K_C \circ K_C \circ \ldots \circ K_C}_{j \text{ times}}$

- **Variance Reduction** then

$$\mathbb{E}_{\pi_F}(Q_F) = \mathbb{E}_{\tilde{\pi}_C}(Q_C) + [\mathbb{E}_{\pi_F}(Q_F) - \mathbb{E}_{\tilde{\pi}_C}(Q_C)]$$

Adaptive Correction - Wrong models can be made less wrong!

- Significant issue if big difference between fine and coarse posterior!



- Every time we do accept / reject we can evaluate $\mathcal{F}_F - \mathcal{F}_C$
- Multilevel trick on our statistical model

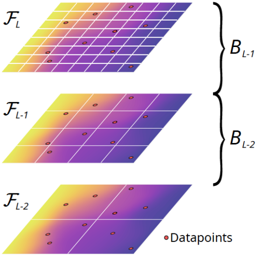
$$\mathbf{d} - \mathcal{F}_C = \underbrace{\mathcal{F}_F - \mathcal{F}_C}_{\mathcal{B}_F \sim \mathcal{N}(\mu_{B,F}, \Sigma_{B,F})} + \underbrace{\mathbf{e}}_{\mathcal{N}(0, \Sigma_e)}$$

Adaptive Correction - Learning on-the-fly

- Likelihood on $\ell - 1$ now addition of two Gaussians

$$\mathcal{L}_C = \exp\left(-\frac{1}{2}(\mathcal{F}_C(\theta) + \mu_{\mathcal{B},F} - \mathbf{d})^{\mathbf{T}}(\boldsymbol{\Sigma}_{\mathcal{B},\mathbf{F}} + \boldsymbol{\Sigma}_{\mathbf{e}})^{-\mathbf{1}}(\mathcal{F}_{\mathbf{C}}(\theta) + \mu_{\mathcal{B},\mathbf{F}} - \mathbf{d})\right)$$

- Repeat over all levels - by summing all biases between levels
- These corrections can be built recursively - little overhead



$$\mu_{F,i+1} = \frac{1}{i+1}\left(i\mu_{F,i} + \mathcal{B}(\theta^{i+1})\right)$$

and

$$\Sigma_{F,i+1} = \frac{i-1}{i}\Sigma_{F,i} + \frac{1}{i}\left(i\mu_{F,i}\mu_{F,i}^T - (i+1)\mu_{F,i+1}\mu_{F,i+1}^T + \mathcal{B}_F(\theta^{i+1})\mathcal{B}_F(\theta^{i+1})^T\right)$$

Open Question: Can you prove adaptive version gives convergence algorithm - without using diminishing adaptivity?

Implementation in `pymc3` - version >3.10

`https://docs.pymc.io`

`https://docs.pymc.io/notebooks/MLDAintroduction.html`

Lightweight code called tinyDA by Mikkel

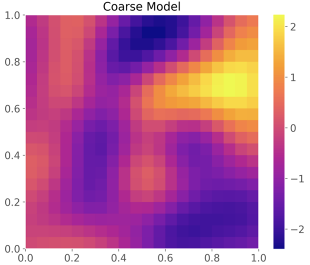
`https://github.com/mikkelbue/tinyDA`

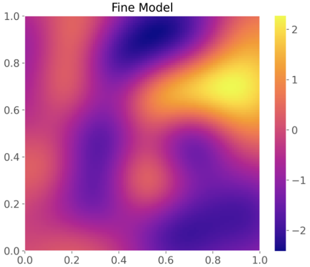
Subsurface flow in Heterogeneous Rock

- Spatially **uncertainty in rock permeability**
  - Parameterised by $\boldsymbol{\theta} \in \mathbb{R}^Z$, Z large $> 1,000$.
- **Sparse measurements** of ‘real’ pressure head
  - Evaluated at $\mathbf{x}^{(j)} \in D$ for $j = 1 \dots M$ points
  - Store in vector $\mathbf{d} \in \mathbb{R}^M$.
- **Forward Model** $\mathcal{F}(\boldsymbol{\theta}) : \mathbb{R}^Z \mapsto \mathbb{R}^M$ predicts pressure at $\mathbf{x}^{(j)}$ given $\boldsymbol{\theta}$.
- **Quantity of Interest** $\mathcal{Q}(\boldsymbol{\theta})$ Could be $\boldsymbol{\theta}$ it's self, full pressure field, flow over boundary
- Introduce a **Gaussian Model** connecting model and data

$$\mathbf{d} = \mathcal{F}(\boldsymbol{\theta}) + \epsilon \quad \text{where} \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \sigma_f^2 \mathbb{I})$$

- Likelihood $\mathcal{L}(\mathbf{d}|\theta) \sim \mathcal{N}(\mathbf{d} - \mathcal{F}(\theta), \sigma_f^2 \mathbb{I})$



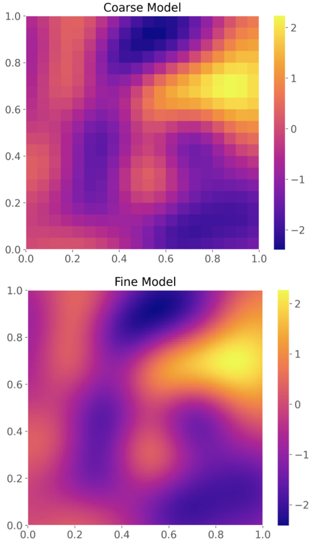


- Our model is **Darcy's equation**.
- $u(\mathbf{x})$ pressure head, $k(\mathbf{x}, \boldsymbol{\theta})$ permeability, source/pumping $f(\mathbf{x})$
- Classical FEM approximation of Darcy equations $u(\mathbf{x}) = \sum_{i=1}^{N} u_i \phi_i(\mathbf{x})$ on $\mathcal{T}_h$, for all $v \in V_h$

$$\int_D k(\mathbf{x}, \boldsymbol{\theta}) \nabla u \cdot \nabla v \, d\mathbf{x} + \int_D f v \, d\mathbf{x} = 0$$

- Large sparse (linear) system of equation

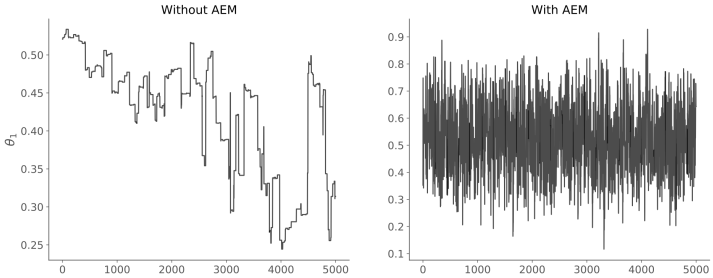
$$\mathbf{A}(\boldsymbol{\theta})\mathbf{u} = \mathbf{b}, \quad \mathbf{u} \in \mathbb{R}^N$$



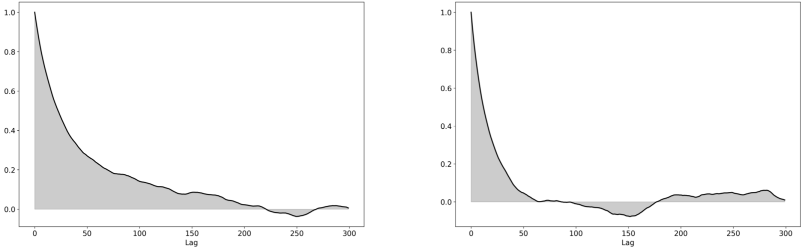
UQ: More independent samples is better!

We sampled the same model, with and without the AEM.

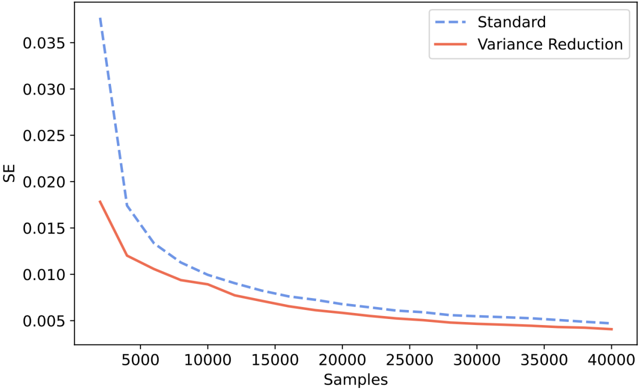
- **Without** AEM:
  - Acceptance rate: 0.02
  - Effective Sample Size, θ_1: 4/20000
- **With** AEM:
  - Acceptance rate: 0.66
  - Effective Sample Size, θ_1: 3319/20000
- 800 fold increase in efficiency.



UQ: More independent samples is better!



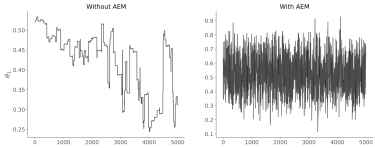
Variance Reduction - nothing to write home about



- Leads to factor ~ 2.5 additional speed up in estimating QoI

Concluding Remarks

- **Addressed Issues** with MLMCMC
- Adaptive Error Model has **large gains**
- Adaptive Multilevel Delayed Acceptance embedded in `pymc3`
- Model hierarchy can be general!
- **Parallelisation** a new challenge
- **New applications** Crystal Plasticity, Fusion Reactor, Trajectory Prediction and Reinforcement Learning . .



MB Lykkegaard, G Mingas, R Scheichl, C Fox, TJ Dodwell, **Multilevel Delayed Acceptance MCMC with an Adaptive Error Model in PyMC3**, NeurIPS, 2020.

MB Lykkegaard, TJ Dodwell, C Fox, R Scheichl**Multilevel Delayed Acceptance MCMC**, *submitted to SIAM JUQ*, Feb 2022.

Thoughts on Parallelisation.

Hedge or Bet?

Formulate as a multi-armed bandit problem using on the fly expected costs and acceptance rates.

Could be more complex $\rightarrow$ state dependent acceptance rates - probably means RL - overkill in my opinion.

Potential if you can use ’transfer’ learning from similar problems.