A Fast Particle-Based Approach for Computer Model Calibration

Murali Haran[1]

Joint work with Ben Seiyon Lee[1], Robert Fuller[2], David Pollard[3] and Klaus Keller[2]

[1]The Pennsylvania State University, Department of Statistics
[2]The Pennsylvania State University, Department of Geo-sciences
[3]The Pennsylvania State University, Earth and Environmental Systems Institute

Joint Statistical Meetings 2018

July 31, 2018

Scientific Motivation

Scientific Question

How much will the **Antarctic Ice Sheet (AIS)** contribute to sea level change?

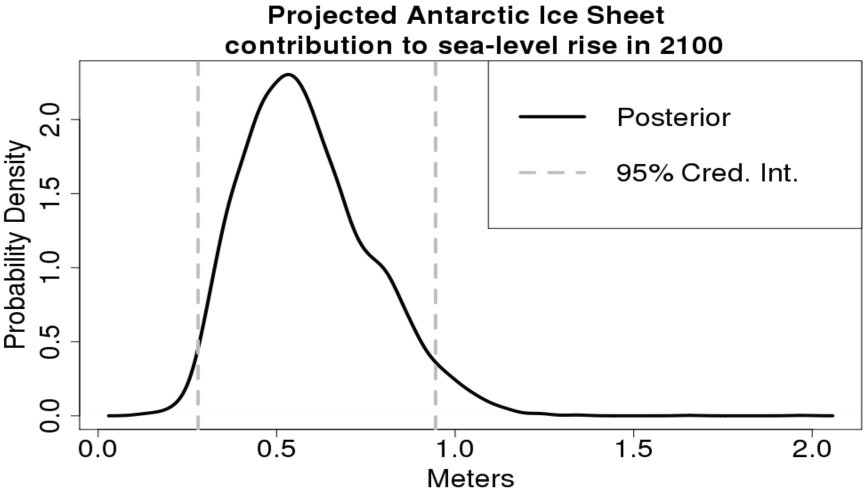
- The Antarctic ice sheet is the single largest source of uncertainty in future sea-level rise (Pollard and DeConto, 2009)
- Ice sheet models provide projections
  e.g DAIS (Shaffer, 2014) and PSU3D-ICE (Pollard and DeConto, 2009)

Statistical Goals

1. Use observations + model outputs to quantify parametric uncertainty
2. Project future contribution to sea level rise, with uncertainty
   (Fuller et al. [2017], Ruckert et al. [2017], and Wong et al. [2017])

End Result

Projection with Uncertainty



This Talk

Background:

- Computer models are often used to make climate projections
  e.g. contribution of Antarctic Ice Sheet to sea level change in 2100
- Key model parameters ("inputs") are often uncertain
  e.g. γ: the sensitivity of ice flow to sea level

Calibration: Estimate parameters using model outputs and observations

Current calibration methods: what they are best suited for

- **Markov Chain Monte Carlo (MCMC):**
  Fast models ($\sim$1-10 seconds per run) and many parameters (5+)
- **Gaussian Process Emulation:**
  Slow models ($\sim$5 mins - many days per run) and few parameters (1-5)

Our contribution:
Fast particle-based approach for computer model calibration useful for moderately fast models (1-5 mins per run) and many parameters (5-20)

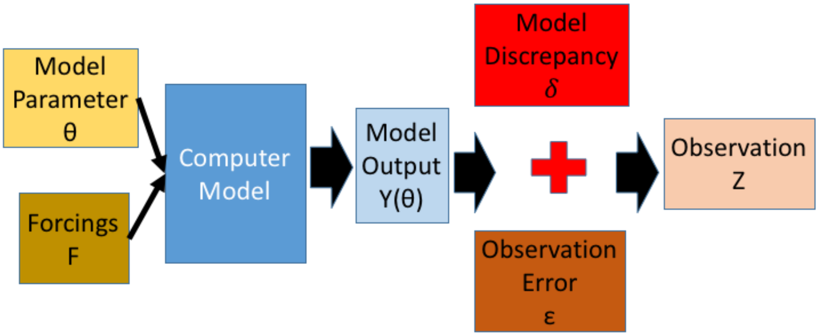
Outline

1. Current Calibration Methods
2. New Particle-based Methods for Calibration
3. Application to Antarctic Ice Sheet Model

Outline

1. Current Calibration Methods
2. New Particle-based Methods for Calibration
3. Application to Antarctic Ice Sheet Model

Computer Model Calibration



Observation Model:

$$Z = Y(\theta) + \delta + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_z^2 I),$$

- **Forcings (F):** Factors that affect the climate system (e.g. global mean temperature)
- **Model Discrepancy (δ):** Systematic difference between observations and model output around the "best" parameter settings

Current Calibration Methods (Kennedy & O'Hagan, 2001)

Observational Model:

$$Z = Y(\theta) + \delta + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_z^2 I),$$

Statistical Inference:

Infer parameters $\theta \in \mathbb{R}^p$ based on posterior distribution, $\pi(\theta, \delta, \sigma_z^2 | Z, Y)$:

$$\boldsymbol{\pi(\theta, \delta, \sigma_z^2 | Z, Y) \propto \underbrace{\mathcal{L}(\theta | Z, Y, \delta)}_{\text{Likelihood}} \times \underbrace{p(\theta, \delta, \sigma_z^2)}_{\text{Priors for } \theta,\ \delta, \text{ and } \sigma_z^2}}$$

Challenge due to slow model runs and many parameters:

- **MCMC:** Expensive computer model runs, $Y(\theta)$, make MCMC computationally prohibitive.
- **Gaussian Process Emulator:**
  Not feasible for high-dimensional parameter spaces.

Outline

Sampling Importance Re-sampling

Goal: Sample from target distribution $\pi(\theta)$
Idea: Sample from distribution $q(\theta)$ + re-sample according to target $\pi(\theta)$
Importance Sampling:

$$E_\pi\left[g(\theta)\right] = E_q\left[g(\theta)\frac{\pi(\theta)}{q(\theta)}\right] = E_q\left[g(\theta)w(\theta)\right]$$

for $q(\cdot)$ s.t. $\pi(\theta) > 0 \Rightarrow q(\theta) > 0, \quad \forall\theta \in \Theta$

Monte Carlo approximation:

1. Simulate $\theta^{(1)}, ..., \theta^{(J)} \sim q(\cdot)$ and generate weights $w^{(j)}$ such that:

$$\frac{1}{J}\sum_{j=1}^{J} w^{(j)} g(\theta^{(j)}) \rightarrow E_\pi\left[g(\theta)\right]$$

2. Resample $\theta_1, ..., \theta_J \sim \{\theta_1, ..., \theta_J\}$ with weights $\{w^{(1)}, ..., w^{(J)}\}$

End Result: Resampled Empirical Target Distribution:

$$\hat{\pi}(\theta) = \frac{1}{J}\sum_{j=1}^{J} \delta_{\theta^{(j)}}(\theta) \approx \pi(\theta)$$

Issues with Sampling Importance Re-Sampling

1. Choice of Importance Function

- **Challenge:** Difficult to find suitable importance distributions
- **Idea:** Sequential Sampling Importance Re-sampling (Chopin [2002]) + Employ intermediate posterior distributions as importance distributions

Intermediate Posterior Distribution:

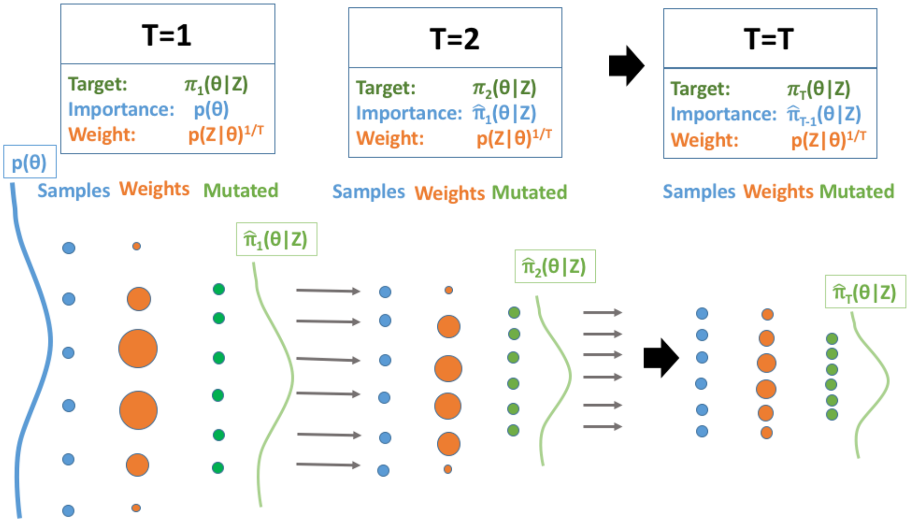
$$\pi_t(\theta|Z) \propto p(Z|\theta)^{t/T} p(\theta)$$

In cycle t, $\pi_t(\theta|Z)$ is the target and $\pi_{t-1}(\theta|Z)$ is the importance distribution.

2. Sample Impoverishment

- **Challenge:** Few unique samples remain after re-sampling
- **Idea:** "Mutate" each particle via Metropolis-Hastings updates (Gilks and Berzuini, 2001)
  - For $j = 1, \ldots J$, set $\theta^{(j)}$ as the initial value, and draw K samples from $\pi(\theta|Z)$ via Metropolis-Hastings
  - "Mutated" $\theta^{(j)} = K$-th sample drawn

Sequential Sampling Importance Resampling + Mutation
Chopin (2002) and Del Moral et al. (2006)



Our Fast Particle-Based Calibration Approach

Outline:

1. Sequential sampling importance resampling
2. Use intermediate posteriors as importance distribution
3. Combat sample impoverishment via Metropolis-Hastings 'mutation'
4. **Stopping Rule:** At each cycle, t:
    1. Obtain samples from the posterior distribution of a **target metric,** $h(\theta_t)$**,** using particles (θ_t).
    2. Calculate the **Bhattacharyya distance** (D_b) between the samples from current cycle t and previous cycle $t-1$.
    3. Stop when $D_b < \epsilon$, where ϵ is a chosen threshold.

Our approach is designed for:

- Computer models with moderate run time ($\sim 1 - 5$ mins)
- $\sim 5 - 20$ model parameters
- High performance computing (parallelization)

Calibration Algorithm

- **Initialization:**
  Draw $\theta_0^{(j)} \sim p(\theta)$ for particles $i = 1, ..., J$. Set weights $w_0^{(i)} = 1/J$
- For cycles $t = 1, ..., T$:
  1. **Compute Importance Weights:** $w_t^{(j)} \propto w_{t-1}^{(j)} \times p(Z_t|\theta_{t-1}^{(j)})$
  2. **Resample:** Draw $\theta_t^{(j)}$ from $\{\theta_{t-1}^{(1)}, ..., \theta_{t-1}^{(J)}\}$ with probabilities $\propto \{w_t^{(1)}, ..., w_t^{(J)}\}$
  3. **Mutation:** Using each particle $(\theta_t^{(1)}, ..., \theta_t^{(J)})$ as an initial value, run J chains of an MCMC algorithm with target $\pi_t(\theta|Z)$
     Set $\theta^{(j)} = K$-th sample drawn from the j-th chain.
  4. **Check stopping criterion**
  5. **If Stopping Criterion achieved:**
     - End Algorithm

     Else:
     - **Reset weights:** $w_t^{(i)} = 1/J$ for particles $i = 1, ..., J$
     - Set t=t+1 and return to Step 1.

Outline

1. Current Calibration Methods
2. New Particle-based Methods for Calibration
3. Application to Antarctic Ice Sheet Model

Application to DAIS Model (Shaffer, 2014):

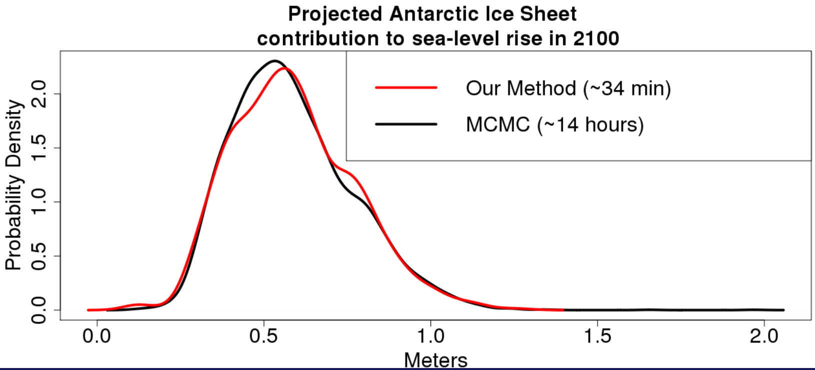
Danish Center for Earth System Science Antarctic Ice Sheet (DAIS) Model

- **Inputs:**
  - 16 model parameters (θ)
  - 3 forcings: Reconstructed sea level, high latitude subsurface ocean temperature, and Antarctic temperature taken at sea-level
- **Observations (Z)**:
  - **Expert Assessments of Antarctic Ice Sheet contribution to sea-level rise**: Last Interglacial Age (118kyr), Last Interglacial Maximum (20kyr), and Mid-Holocene (6kyr)
  - **Change in Antarctic Ice Sheet contribution to sea-level rise** 1993-2010, 1992-2001, and 2002-2011 (Intergovernmental Panel on Climate Change [IPCC])
- **Computer model runs** $\left[Y(\theta)\right]$
  Computer model output corresponding to above observations

Goal: Infer model parameters, θ.

DAIS Model Calibration Results

- J=1000 particles and K=200 Metropolis-Hastings updates per cycle
- Comparable calibration results to those from MCMC-based methods
- **~34 minutes (Our Approach)** vs. **14 hours (MCMC [Gold Standard])**
- Our method will allow us to calibrate PSU3D-ICE using Pliocene data
  MCMC: ≈ 5 years vs. Particle-based Approach: ≈ 10 days



Acknowledgements

This work was supported by the National Science Foundation through the **Network for Sustainable Climate Risk Management (SCRiM) under NSF cooperative agreement GEO-1240507** and by the US Department of Energy, Office of Science through the **Program on Coupled Human and Earth Systems (PCHES) under DOE Cooperative Agreement No. DE-SC0016162**.

References

[1] Chopin, N. (2002). A sequential particle filter method for static models. *Biometrika*, 89(3):539–552.

[2] Crisan, D. and Doucet, A. (2000). Convergence of sequential Monte Carlo methods. *Signal Processing Group, Department of Engineering, University of Cambridge, Technical Report CUEDIF-INFENGrrR38*, 1.

[3] DeConto, R. M. and Pollard, D. (2016). Contribution of Antarctica to past and future sea-level rise. *Nature*, 531(7596):591.

[4] Del Moral, P., Doucet, A., and Jasra, A. (2006). Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B*, 68(3):411–436.

[5] Fuller, R. W., Wong, T. E., and Keller, K. (2017). Probabilistic inversion of expert assessments to inform projections about Antarctic ice sheet responses. *PloS One*, 12(12):e0190115.

[6] Kennedy, M. C. and O'Hagan, A. (2001). Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B*, 63(3):425–464.

[7] Ruckert, K. L., Shaffer, G., Pollard, D., Guan, Y., Wong, T. E., Forest, C. E., and Keller, K. (2017). Assessing the impact of retreat mechanisms in a simple Antarctic ice sheet model using Bayesian calibration. *PloS One*, 12(1):e0170052.

[8] Shaffer, G. (2014). Formulation, calibration and validation of the DAIS model, a simple antarctic ice sheet model sensitive to variations of sea level and ocean subsurface temperature. *Geoscientific Model Development*, 7(4):1803–1818.

Thank you!

Appendix

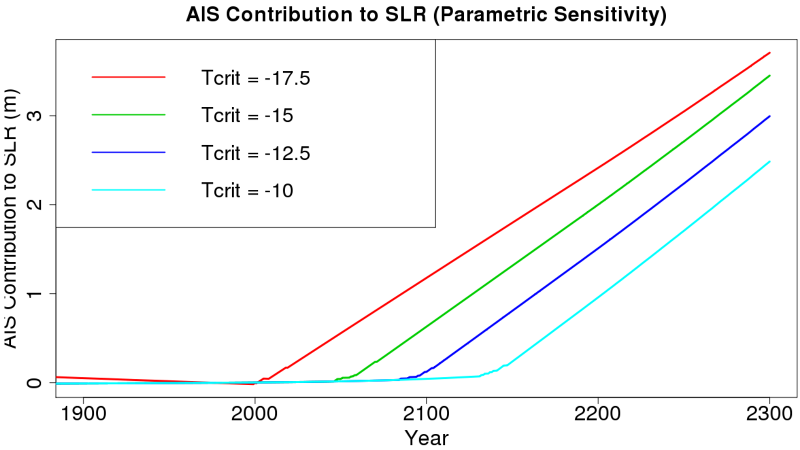
DAIS (Fast Dynamics) Model Components and Output:

DAIS (Antarctic Ice Sheet)

- **Input:** 15 model parameters
  1. a_{anto}: Sensitivity of Antarctic Ocean temp. to global mean surf. temp.
  2. b_{anto}: Approximate ocean temperature
  3. γ: Sensitivity of ice flow to sea level
  4. α: Sensitivity of ice flow to ocean subsurface temperature
  5. μ: Profile parameter related to ice stress
  6. ν: Relates balance gradient to precipitation
  7. P_0: Precipitation at 0°C
  8. κ: Relates precipitation to temperature
  9. F_O: Constant of proportionality for ice speed
  10. H_0: Initial value for runoff line calculation
  11. c: Second value for runoff line calculation
  12. B_0: Height of bed at the center of the continent
  13. slope: Slope of the bed
  14. T_{crit}: Threshold temperature for fast disintegration of AIS
  15. λ: Fast dynamical ice sheet disintegration rate
- **Output:** Antarctic ice sheet volume

Importance of Parametric Uncertainty

Example Parameter for DAIS model: T_{crit}.
When Antarctic sea-level temperature rises above T_{crit}, it triggers fast disintegration of the Antarctic Ice Sheet.



MCMC (Gold Standard)

Observation Model:

$$Z = Y(\theta) + \delta + \epsilon, \qquad \epsilon \sim \mathcal{N}(0, \sigma_z^2 I),$$

- Model Output at input θ: $Y(\theta) \in \mathbb{R}^n$
- Model Parameters: $\theta \in \Theta$
- Observations: $Z \in \mathbb{R}^n$
- Measurement error variance: σ_z^2
- Discrepancy term: δ

Statistical Inference:
Infer parameters θ based on $\pi(\theta, \delta, \sigma_z^2 | Z, Y)$, where:

$$\boldsymbol{\pi(\theta, \delta, \sigma_z^2 | Z, Y) \propto \underbrace{\mathcal{L}(\theta, \sigma_z^2 | Z, Y, \delta)}_{\text{Likelihood}} \times \underbrace{p(\theta) p(\delta) p(\sigma_z^2)}_{\text{Apriori Independent Priors}}}$$

Challenge: Costly to draw samples from $\pi(\theta, \delta, \sigma_z^2 | Z, Y)$ via MCMC.

Calibration with Emulation

Main Idea: Replace computer model with Gaussian Process Emulator.

Step #1: Emulation

- Fit a zero-mean Gaussian Process on a design set of inputs and outputs.
- GPE generates approximate outputs ($\eta(\theta_j)$) at untried input settings (θ_j)

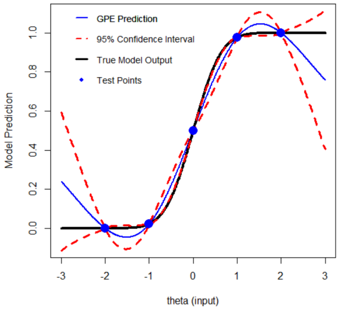
Step #2: Calibration

- Replace model output $Y(\theta_j)$'s with GPE output $\eta(\theta_j)$.
- Infer model parameters based on the posterior distribution.

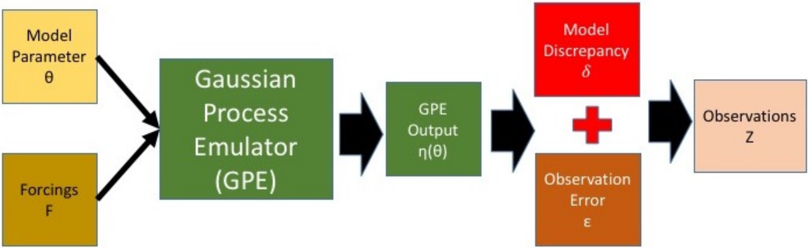
Challenge:

- Not feasible for high-dimensional parameter spaces.
- Still requires MCMC.

GPE vs. True Model



Computer Model Calibration with Gaussian Process Emulator

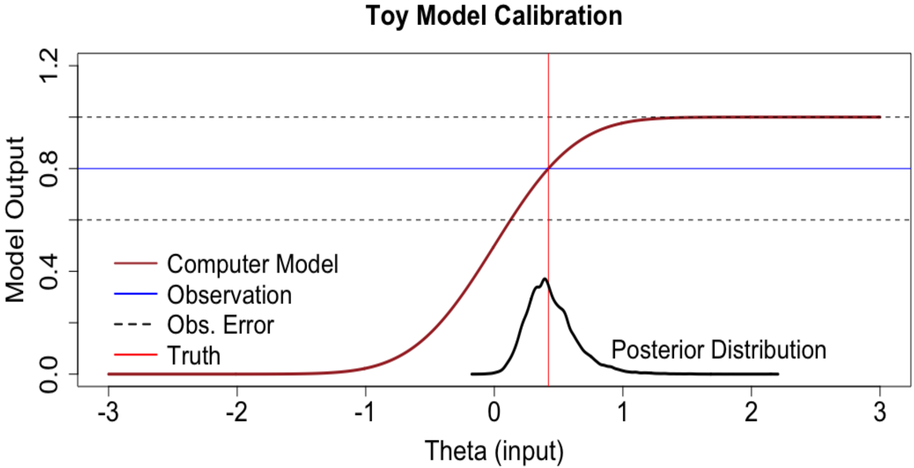


Observational Model:

$$Z = \eta(\theta) + \delta + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_z^2 I),$$

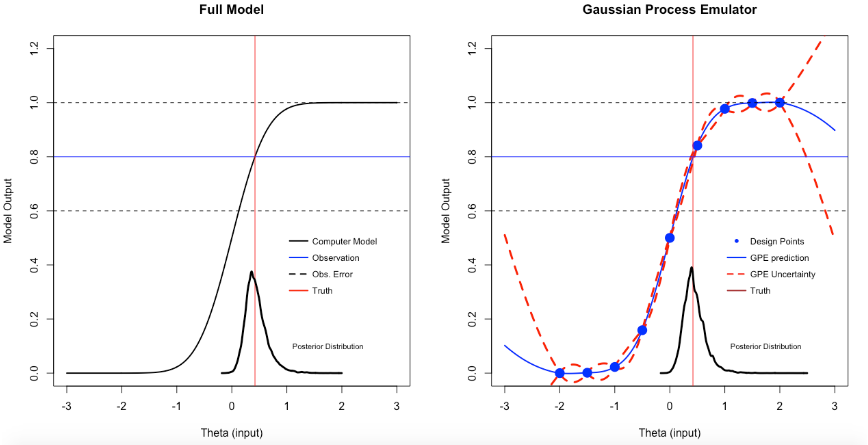
Calibration Example:

Objective:
What is θ given **observations** and **computer model runs**?



Calibration Example:

Objective:
[LEFT] What is θ given obs. (horizontal blue) and computer model runs (black)?
[RIGHT] What is θ given obs. (horizontal blue) and GPE runs (blue)?



Tempering Methods for Static Problems (Chopin, 2007)

Static Computer Model: Generates entire time series ($Y_{1:n}$) at once.
Problem: Can run one instance of importance sampling.

Solution: Turn a static problem into a sequential one via tempering.

$\boxed{\textbf{Method 1:}}$

Data Tempering: Partition data into J subsets and assimilate each subset sequentially.

$$\pi_{n_J}(\theta|Z_{0:n_J}) \propto p(\theta) \times L(\theta|Z_{0:n_J})$$

$$n_0 = 0 < n_1 < \cdots < n_J = n$$

$\boxed{\textbf{Method 2:}}$

Likelihood Tempering: Assimilate J copies of the data. Likelihood at each iteration is a fractional power of the full likelihood.

$$\pi_{\gamma_J}(\theta|Z) \propto p(\theta) \times p(Z|\theta)^{\gamma_t}$$

$$\gamma_0 = 0 < \gamma_1 < \cdots < \gamma_J = 1$$

Better Importance Distributions (Intermediate Posterior Distributions)

Challenge: Difficult to find suitable importance distributions

Idea: Generate intermediate posterior distributions (similar) to act as importance distributions

Tempered Likelihood (Chopin, 2002):

$$p(Z|\theta) = \prod^{T} p(Z|\theta)^{1/T}$$

Posterior Distribution

$$\pi(\theta|Z) \propto \left[\prod^{T} p(Z|\theta)^{1/T}\right] p(\theta)$$

$\boxed{\textbf{Key Result:}}$ **Intermediate Posterior Distribution**

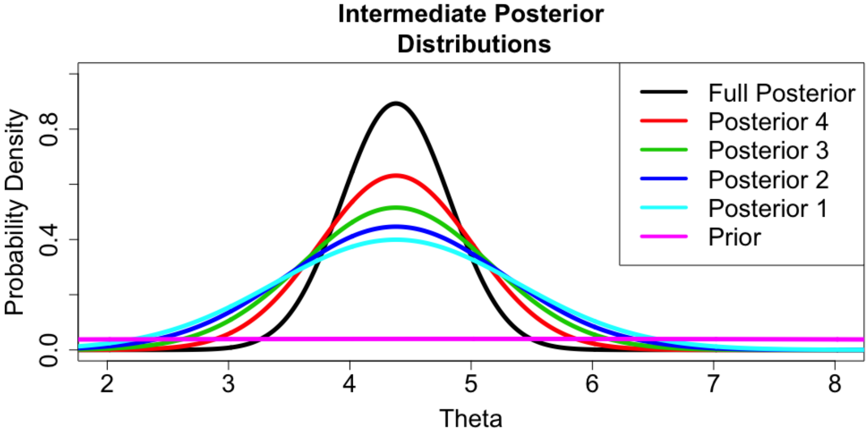
For cycle $t = 1, ..., T$, we have:

$$\pi_t(\theta|Z) \propto p(Z|\theta)^{t/T} p(\theta)$$

Sequential Sampling Importance Resampling

In cycle t:

1. **Target Distribution:** $\pi_t(\theta|Z) \propto p(Z|\theta)^{t/T} p(\theta)$
2. **Importance Distribution:** $\pi_{t-1}(\theta|Z) \propto p(Z|\theta)^{t-1/T} p(\theta)$



Intermediate Posterior Distributions

Weight at cycle t

$$w_t^{(j)} = \frac{\pi_t(\theta|Z)}{\pi_{t-1}(\theta|Z)} \propto \frac{p(Z|\theta)^{t/T}p(\theta)}{p(Z|\theta)^{(t-1)/T}p(\theta)} = p(Z|\theta)^{1/T}$$

Sampling Importance Resampling with Mutation

Sample Impoverishment

Problem: Few unique samples remain after resampling

Solution: "Mutate" particles via Metropolis-Hastings updates (Gilks and Berzuini, 2001)

- For $j = 1, ...J$, set $\theta^{(j)}$ as the initial value, and draw K samples from $\pi(\theta|Z)$ via Metropolis-Hastings
- "Mutated" $\theta^{(j)} = K$-th sample drawn
- Note: K is small and we run J separate Markov chains

Bhattacharyya Distance

Idea: Measure similarity between two distributions (p,q) by examining the overlap between the corresponding samples

Bhattacharyya Distance (D_B): Measures the overlap between two samples. First, split/partition the two samples into n partition intervals. Then,

$$D_B(p,q) = -\ln\left(\sum_{i=1}^{n}\sqrt{p_i q_i}\right)$$

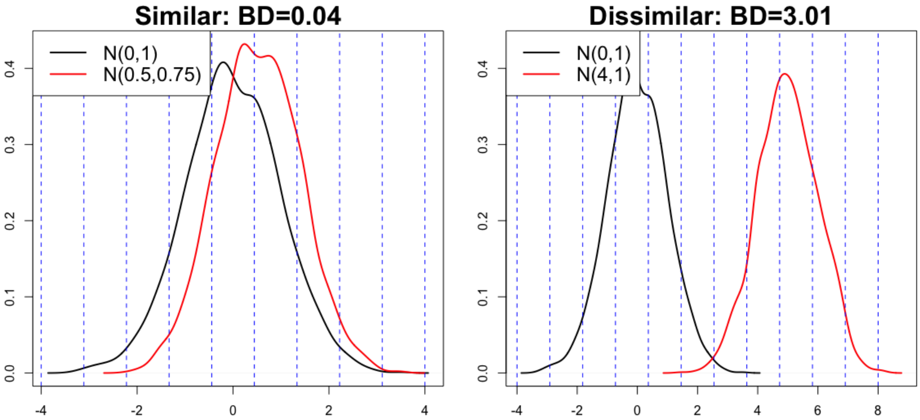
where p_i and q_i are the **proportion** of samples within the **i-th** partition, and $D_B \in \mathbb{R}^+$.

Result:

1. Small $D_B \Rightarrow$ Lot of overlap between samples (Similar)
2. Large $D_B \Rightarrow$ Little overlap between samples (Not Similar)

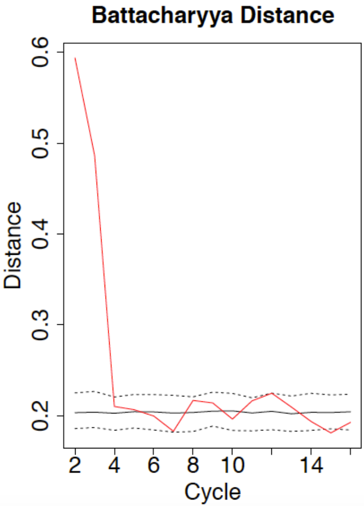
Example of Bhattacharyya Distance

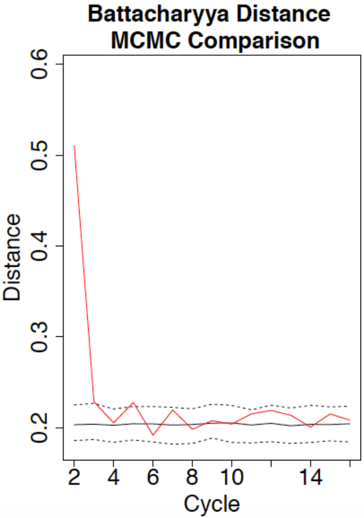
Similar samples yield smaller Bhattacharyya distances



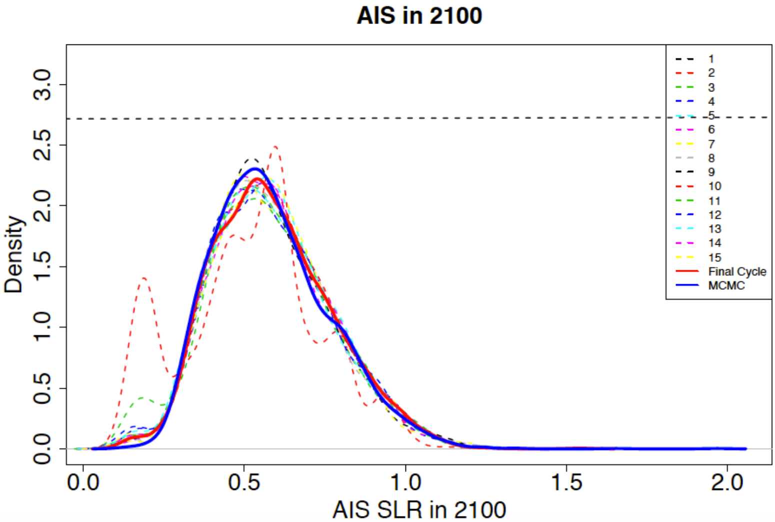
Stopping Criterion for DAIS Model

Run algorithm until Bhattacharyya Distance dips below range (dotted black lines). Range generated by bootstrapping.

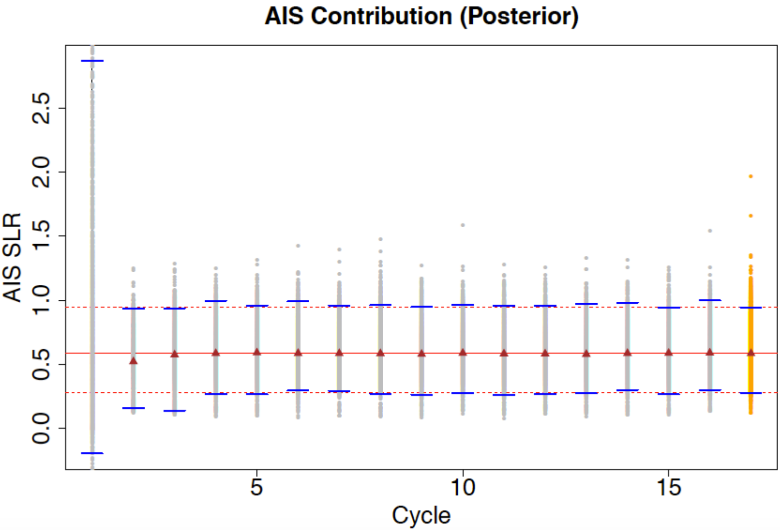




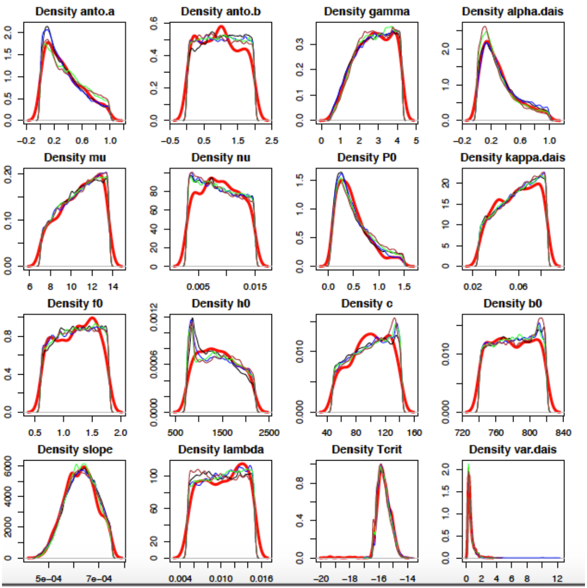
Antarctic Ice Sheet Contribution to Sea Level Rise By Cycle



Antarctic Ice Sheet Contribution to Sea Level Rise (Particles)



DAIS Model Parameters Calibration Results



Convergence of Empirical Distributions (Crisan and Doucet, 2000)

Empirical Distributions

1. **Prior Distribution:** For $\theta^{(1)}, ..., \theta^{(J)} \sim p(\theta)$,

$$\hat{p}(\theta) = \frac{1}{J}\sum_{j=1}^{J}\delta_{\theta^{(j)}}(\theta)$$

2. **Intermediate Posterior Distribution:**
Based on the "mutated" samples $\theta^{(1)}, ..., \theta^{(J)}$ after cycle t,

$$\hat{\pi}_t(\theta|Z) = \frac{1}{J}\sum_{j=1}^{J}\delta_{\theta^{(j)}}(\theta)$$

Theorem

Observing that, almost surely, $\lim_{N\to\infty}\hat{p} = p$*, for all* $t \geq 0$*, we have, almost surely,* $\lim_{N\to\infty}\hat{\pi}_t(\theta|Z) = \pi_t(\theta|Z)$

PF with Single Resampling Move (Gaussian Jittering)

Algorithm

1. **Initialization:**
   Draw $\theta_0^i \sim p(\theta)$, $\sigma_0^{2i} \sim p(\sigma^2)$, $\delta_0^i \sim p(\delta)$ for particles $i = 1, ..., M$. Set weights $w_i^0 = 1/M$.

2. For $t = 1, ..., N$ subsets
   1. **Re-weighting:** $w_t^i \propto w_{t-1}^i \times p(Z_t | \theta_{t-1}^i, \sigma_{t-1}^{2i}, \delta_{t-1}^i)$
   2. **Normalize Weights:** $\tilde{w}_t^i = w_t^i / \sum_{k=1}^M w_t^k$
   3. **Re-sampling:** Draw $\theta_t^i, \sigma_t^{2i}, \delta_t^i$ from $\{\theta_{t-1}^i, \sigma_{t-1}^{2i}, \delta_{t-1}^i\}$ with weights $\tilde{w}_t^i$
   4. **Jitter Particles:** Draw $\tilde{\theta}_t^i = (\theta_t^i, \sigma_t^{2i}, \delta_t^i)$ from $\mathcal{N}(\tilde{\theta}_t^i, \tilde{\Sigma}_t)$ where:

   $$\tilde{\Sigma}_t = \frac{\sum_{j=1}^M \tilde{w}_t^j \{\tilde{\theta}_t^j - \hat{E}_t\}\{\tilde{\theta}_t^j - \hat{E}_t\}}{\sum_{j=1}^M \tilde{w}_t^j} \quad \hat{E}_t = \frac{\sum_{j=1}^M \tilde{w}_t^j \tilde{\theta}_t^j}{\sum_{j=1}^M \tilde{w}_t^j}$$

   5. **Reset weights:** $w_t^i = 1/M$ for particles $i = 1, ..., M$.

PF with M-H Transition Kernel (Particle-based)

Algorithm

1. **Initialization:**
   Draw $\theta_0^i \sim p(\theta)$, $\sigma_0^{2i} \sim p(\sigma^2)$, $\delta_0^i \sim p(\delta)$ for particles $i = 1, ..., M$. Set weights $w_i^0 = 1/M$.

2. For $t = 1, ..., N$ subsets
   1. **Re-weighting:** $w_t^i \propto w_{t-1}^i \times p(Z_t|\theta_{t-1}^i, \sigma_{t-1}^{2i}, \delta_{t-1}^i)$
   2. **Normalize Weights:** $\tilde{w}_t^i = w_t^i / \sum_{k=1}^M w_t^k$
   3. **Re-sampling:** Draw $\theta_t^i, \sigma_t^{2i}, \delta_t^i$ from $\{\theta_{t-1}^i, \sigma_{t-1}^{2i}, \delta_{t-1}^i\}$ with weights $\tilde{w}_t^i$
   4. **Move Particles:** For each particle $i = 1, ...M$, run 20-200 iterations of M-H algorithm with proposal distribution $\mathcal{N}(\tilde{\theta}_t^i, \frac{(2.38)^2}{3}\tilde{\Sigma}_t)$ where:

      $$\tilde{\theta}_t^i = (\theta_t^i, \sigma_t^{2i}, \delta_t^i) \quad \tilde{\Sigma}_t = \frac{\sum_{j=1}^M \tilde{w}_t^j \{\tilde{\theta}_t^j - \hat{E}_t\}\{\tilde{\theta}_t^j - \hat{E}_t\}}{\sum_{j=1}^M \tilde{w}_t^j} \quad \hat{E}_t = \frac{\sum_{j=1}^M \tilde{w}_t^j \tilde{\theta}_t^j}{\sum_{j=1}^M \tilde{w}_t^j}$$

   5. **Reset weights:** $w_t^i = 1/M$ for particles $i = 1, ..., M$.

Simulated Example:

Computer Model (Bayarri et al. [2007])

$$Y(\theta) = 5e^{-\mu t}$$

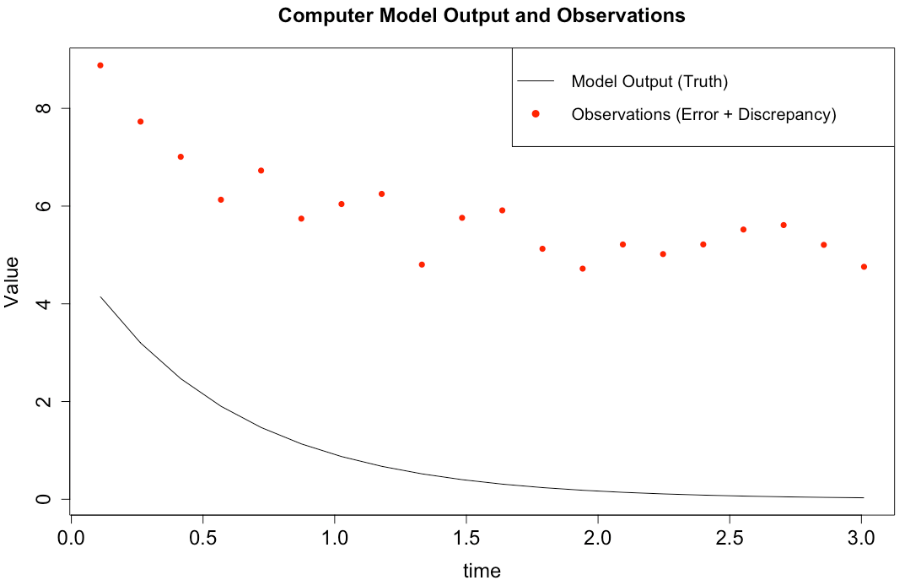
where $\mu = 1.7$ and t are 20 equally spaced points in [0.11,3.01].

Observation Model (Scalar Discrepancy)

$$Z = Y(\theta) + \delta + \epsilon, \qquad \epsilon \sim \mathcal{N}(0, \sigma_z^2 I),$$

where $\delta = 5$ and $\sigma_z^2 = 0.25$.

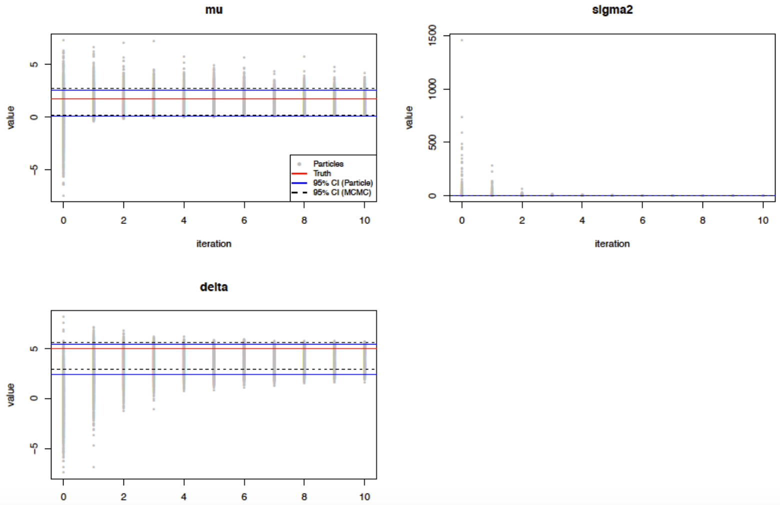
Model Output and Observations



Simulated Examples: Results

Observation Model 2 (Scalar Discrepancy)

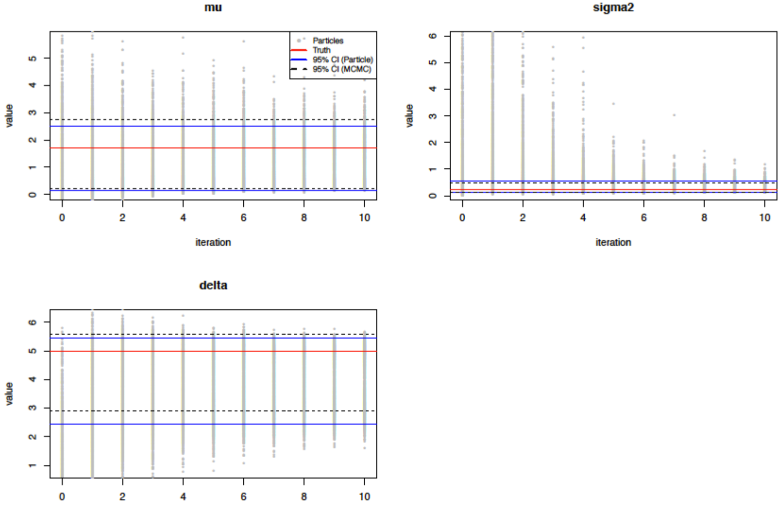
$$Z = Y(\theta) + \delta + \epsilon, \qquad \epsilon \sim \mathcal{N}(0, \sigma_z^2 I),$$



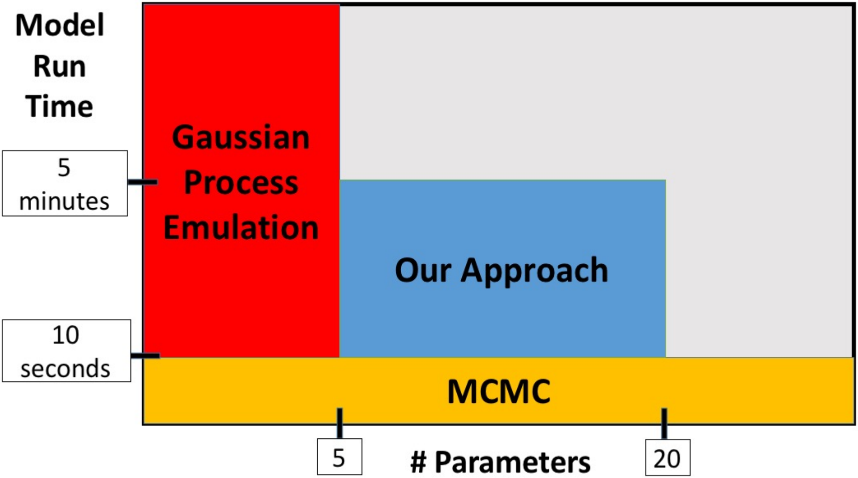
Simulated Examples: Results

Observation Model: (Scalar Discrepancy)

$$Z = Y(\theta) + \delta + \epsilon, \qquad \epsilon \sim \mathcal{N}(0, \sigma_z^2 I),$$



Which Calibration Method Should I Choose?



Parallelization

Initialization:

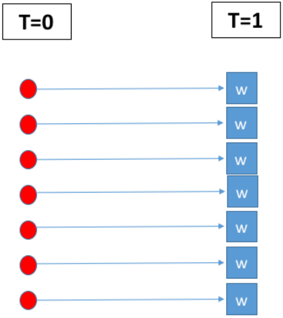
1. **Initialize processors for parallelization**
2. **Draw particles from prior distribution**

T=0

Parallelization

Initialization:
1. Initialize processors for parallelization
2. Draw particles from prior distribution

Parallelization #1:
1. Run model
2. Compute weights



Parallelization

Initialization:

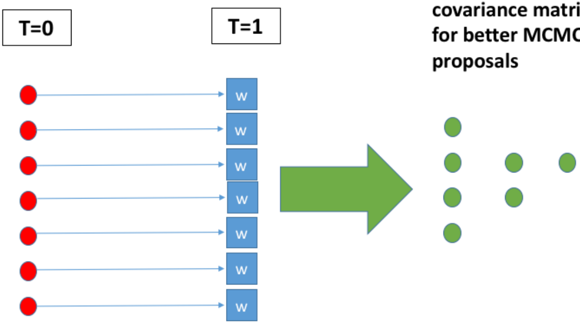
1. Initialize processors for parallelization
2. Draw particles from prior distribution

Parallelization #1:

1. Run model
2. Compute weights

Central Node:

1. Normalize Weights
2. Check Effective Sample Size
3. Resample
4. Compute sample covariance matrix for better MCMC proposals



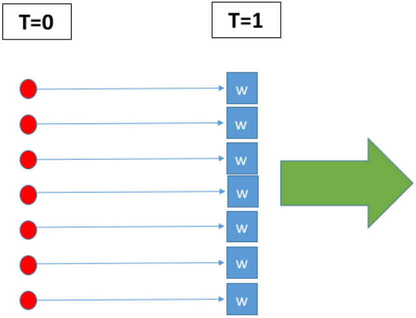
Parallelization

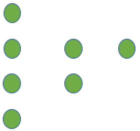
Initialization:
1. Initialize processors for parallelization
2. Draw particles from prior distribution

Parallelization #1:
1. Run model
2. Compute weights

Central Node:
1. Normalize Weights
2. Check Effective Sample Size
3. Resample
4. Compute sample covariance matrix for better MCMC proposals

Parallelization #2:
1. "Mutate" particles via MCMC
2. Repeat Parallelization Part #1 for next cycle





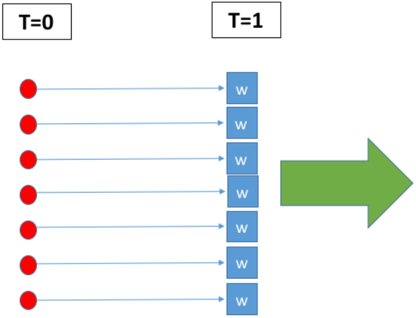
Parallelization

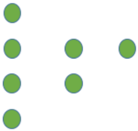
Initialization:
1. Initialize processors for parallelization
2. Draw particles from prior distribution

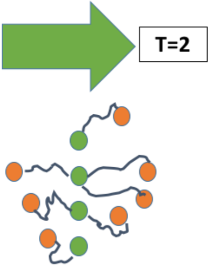
Parallelization #1:
1. Run model
2. Compute weights

Central Node:
1. Normalize Weights
2. Check Effective Sample Size
3. Resample
4. Compute sample covariance matrix for better MCMC proposals

Parallelization #2:
1. "Mutate" particles via MCMC
2. Repeat Parallelization Part #1 for next cycle







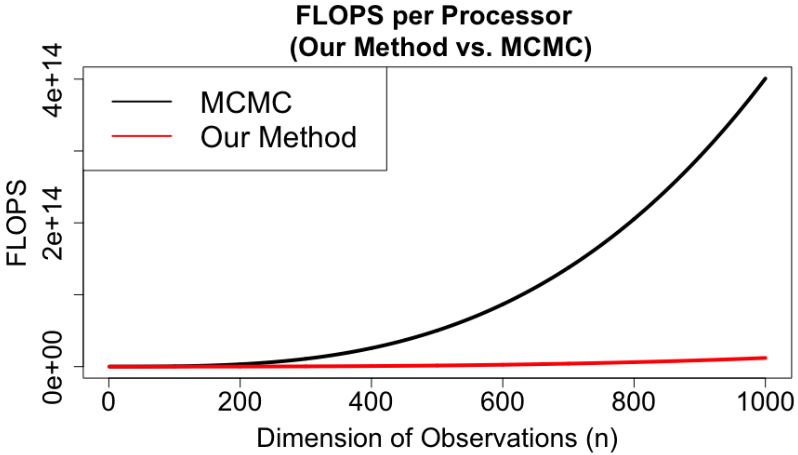
Incorporating Information from the Distant Past

DAIS with observations from the Pliocene ($\sim$2.6-5.3 million years before present)

- Pliocene is a better analog for modern climate than the Last Interglacial Age (118kyr)
- Adding observations from the distant past may provide sharper projections of AIS contribution to sea level change
- Calibration via MCMC projected to take $\sim$ 22 days
- We estimate $\sim$ 5 hours using our method

Conclusions based on DAIS Model

1. Particle-based calibration results comparable to those from MCMC
2. Takes advantage of parallel computing resources (ACI Cluster)
3. Scales well to high-dimensional computer model output



3D Ice Sheet Model

3D Antarctic ice sheet model (PSU3D-ICE)
(collaboration with David Pollard, EESI)

Computer Model:
Pennsylvania State University's three-dimensional ice sheet model, (PSU3D-ICE), simulates the Antarctic Ice Sheet realistically

Scientific Problem:
PSU3D-ICE has not been calibrated using data from the Pliocene ($\sim$ 2.6-5.3 Million Years Before Present). There remains much parametric uncertainty

Research Question: How does calibrating the PSU3D-ICE model with "deep time" observations help inform future flood risks?

Current Work: Calibration via MCMC may take $\sim$ 5 years, but estimate $\sim 5 - 10$ days using our method