PSAAP III
INTEGRATED SIMULATIONS USING EXASCALE MULTIPHYSICS ENSEMBLES

Multifidelity Predictions using Surrogate Models

Elliot Epstein, Dhruv Patel, Eric Darve
Stanford University

I. Motivation

Goal: Given an ignition probability of 0.5,
find probability distribution of input parameters

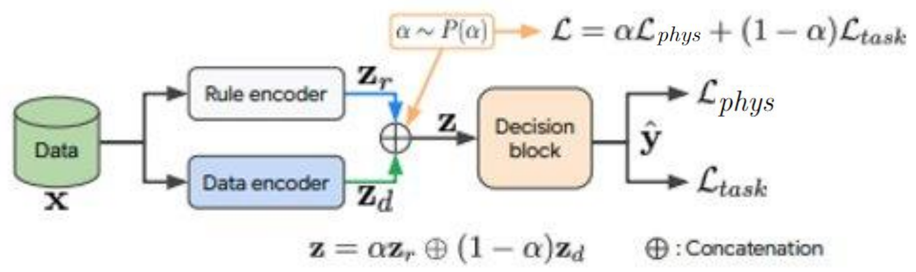
Challenges:

- High fidelity simulations are very expensive
- Sampling using standard MCMC will waste many high-fidelity simulations

Idea: Create a differentiable low fidelity model (with DeepCTRL) and utilize efficient MCMC algorithms that need differentiable input function

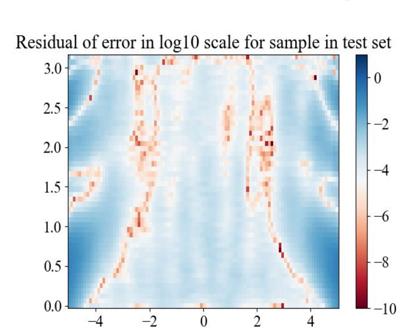
II. DeepCTRL

- A method for learning low fidelity mapping from BC to solution of PDE by learning both from data and physical constraints
- Typical approach: $\mathcal{L} = \mathcal{L}_{task} + \lambda \mathcal{L}_{phys}$
- Novel approach: Use the DeepCTRL loss architecture

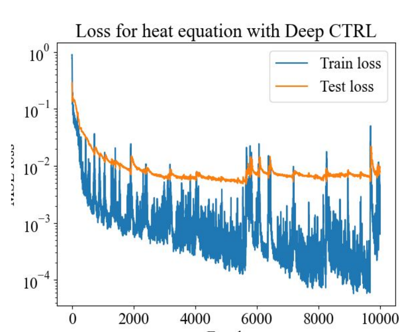


Alpha, the stochastic parameter, is sampled for each data batch and used in both the neural network architecture and loss function

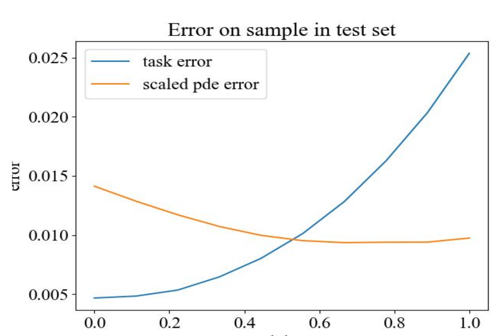
- Setting: Train data is 1000 BC, solution grid pairs, with boundary conditions given by sinusoidal functions with different phase and amplitude. We test the algorithm on the heat equation in 2D.



The error is largest near the domain boundary



The train loss exhibit large stochasticity due to the alpha parameter



At inference, we can pick alpha to either get small task error or small error from physical constraints

III. Multifidelity Hamiltonian Monte Carlo

Typical approach:

$$p_{\mathcal{X}}^{\text{post}}(\boldsymbol{x}|\hat{\boldsymbol{y}}) = \frac{p_{\mathcal{X}}^{\text{like}}(\hat{\boldsymbol{y}}|\boldsymbol{x}) p_{\mathcal{X}}^{\text{prior}}(\boldsymbol{x})}{p_{\mathcal{Y}}(\hat{\boldsymbol{y}})} \propto p_{\eta}(\hat{\boldsymbol{y}} - \boldsymbol{f}(\boldsymbol{x})) p_{\mathcal{X}}^{\text{prior}}(\boldsymbol{x})$$

- Posterior distribution:
- Construct prior, $p_{\mathcal{X}}^{\text{prior}}(\boldsymbol{x})$ (uniform)
- Compute likelihood, $p_{\eta}(\hat{\boldsymbol{y}} - \boldsymbol{f}(\boldsymbol{x}))$ (gaussian)
- Given fixed $\hat{\boldsymbol{y}}$, generate sample from, $p_{\mathcal{X}}^{\text{post}}(\boldsymbol{x}|\hat{\boldsymbol{y}})$, using MCMC with HF model for $\boldsymbol{f}$

Key Idea:

- Use a differentiable LF surrogate model. This allows the use of effective gradient based MCMC methods (HMC). Once a sample is accepted, run the HF model with that input, and accept according to the Metropolis Hastings algorithm

- Synthetic Experiment:

$$HF(x) = \frac{1}{1 + e^{-\alpha_{HF}(\hat{x}^T c)}}$$

$$LF(x) = \frac{1}{1 + e^{-\hat{x}^T c}}$$

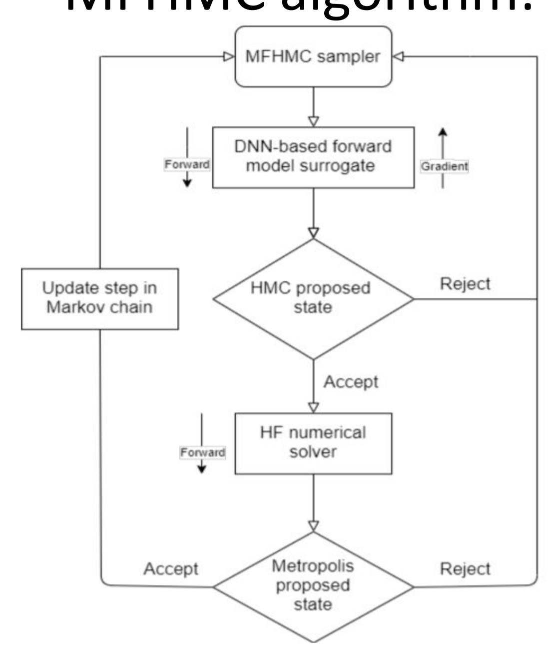
$$\hat{x} = (\sin(x_1), x_2, ..., x_n)^T$$

$$P_{HF}^{post}(x|\hat{y} = 0.5) \propto \mathbb{1}_{x \in C} e^{-\frac{(HF(x)-\hat{y})^2}{2\sigma_{HF}^2}}$$

$$P_{LF}^{post}(x|\hat{y} = 0.5) \propto \mathbb{1}_{x \in C} e^{-\frac{(LF(x)-\hat{y})^2}{2\sigma_{LF}^2}}$$

MH = Metropolis-Hastings algorithm
MCMC = one stage MCMC using MH, LF model is not used
MFMCMC = first stage use MH on LF model, second stage use MH on HF model
MFHMC = first stage use HMC on LF model, second stage use MH on HF model

- MFHMC algorithm:



- Acceptance probabilities in stage 1 and stage 2:

$$\alpha_1 = \min(1, \frac{P_{LF}^{post}(x_1|\hat{y} = 0.5)}{P_{LF}^{post}(x_0|\hat{y} = 0.5)})$$

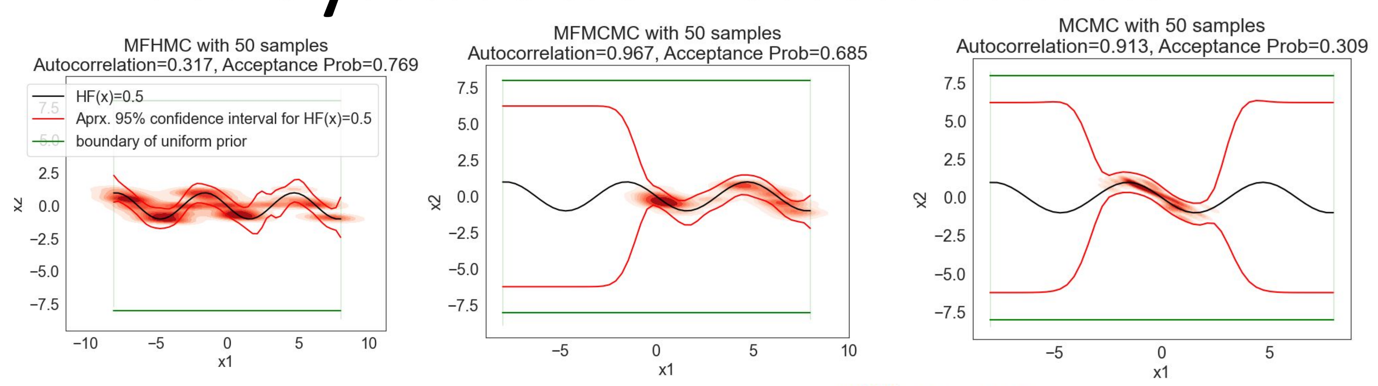
$$\alpha_2 = \min(1, \frac{P_{HF}^{post}(x_1|\hat{y} = 0.5) P_{LF}^{post}(x_0|\hat{y} = 0.5)}{P_{HF}^{post}(x_0|\hat{y} = 0.5) P_{LF}^{post}(x_1|\hat{y} = 0.5)})$$

- Evaluation metrics:

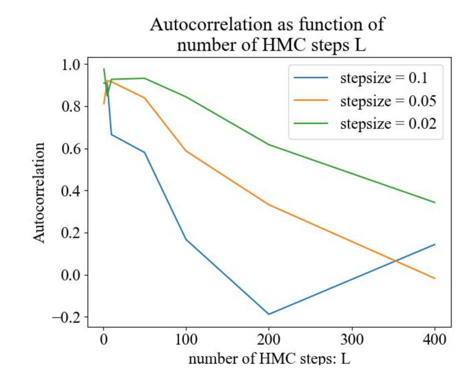
$$\text{Acceptance probability} = \frac{\text{\# collected samples}}{\text{\#HF model evaluations}}$$

$$\text{Autocorrelation} = \text{max of absolute values of autocorrelations along each dim}$$

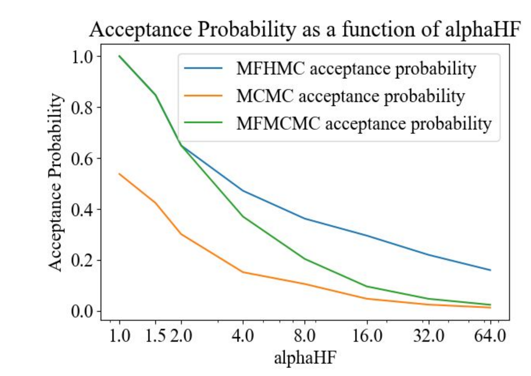
IV. Synthetic Benchmarks



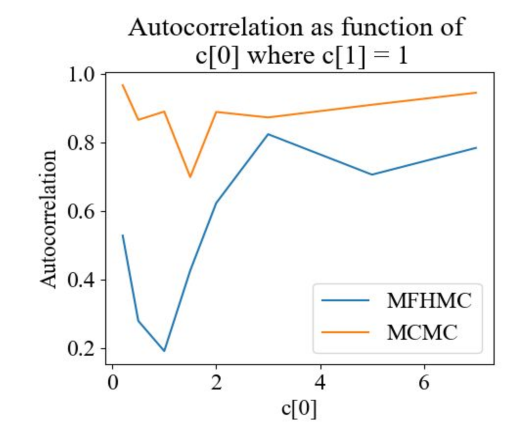
The above figures show the empirical density of $P_{HF}^{post}(x|\hat{y} = 0.5)$, using different sampling algorithms. MFHMC accurately model the full posterior over the region of interest (green box), whereas both MCMC and MFMCMC only sample a subset of the distribution.

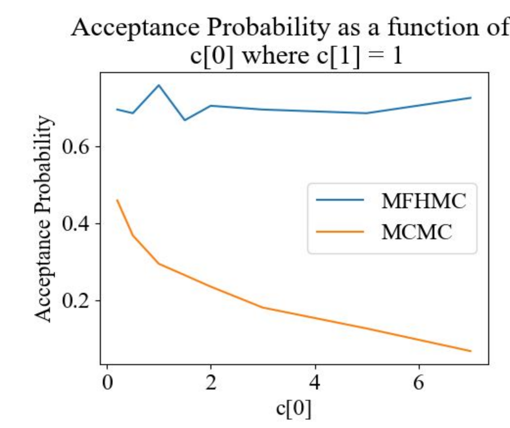


As expected, using a higher numer of HMC steps decreases the autocorrelation for the benchmark



Increasing alphaHF makes the LF and HF models more different, thus increasing alphaHF makes MFHMC less efficient





Increasing c[0] correspond to a sinusoidal decision boundary with increasing amplitude. The autocorrelation for MFHMC increases with c[0], and the acceptance probability remain constant, in contrast to MCMC

V. Ongoing Work

- Implementing the MFHMC algorithm using HTR 2D simulations
- Use as a principled way to run experiments in the PSAAP project
- Datadriven low-fidelity model, in low dimensions based on GP regression and based on neural networks in high dimension
- Active learning to learn the low fidelity model more efficiently

References:

- Sungyong Seo et al. "Controlling Neural Networks with Rule Representations." NeurIPS 2021.
- Dhruv V. Patel. "Multi-Fidelity Hamiltonian Monte Carlo Method with Deep Learning-based Surrogate." AAAI Fall Symposium series (FSS), virtual, 2021.
- Betancourt, Michael, Simon Byrne, Sam Livingstone, and Mark Girolami. "The geometric foundations of Hamiltonian Monte Carlo." Bernoulli 23, no. 4A (2017): 2257-2298.

Contacts: epsteine@stanford.edu, darve@stanford.edu