

Advances in Multi-Fidelity Computer Experiments with Tuning Parameters

Chih-Li Sung

Department of Statistics and Probability
Michigan State University

Seminar, Institute of Statistical Science, Academia Sinica
May 7, 2025



MICHIGAN STATE
UNIVERSITY

Outline

1 Introduction

- Multi-fidelity data
- Auto-Regressive Model

2 Active learning for Finite Element Simulations

3 Diffusion Non-Additive Model

4 Conclusion



Junoh Heo



Romain Boutelet



Chih-Li Sung

Multi-Fidelity Simulations

- Computer models have been widely adopted to understand a real-world feature, phenomenon or event.
- Computer simulations are used to solve these models (e.g., finite element / finite difference) .

Multi-Fidelity Simulations

- Computer models have been widely adopted to understand a real-world feature, phenomenon or event.
- Computer simulations are used to solve these models (e.g., finite element / finite difference) .
- The simulation can be either
 - High-fidelity simulation: costly but close to the truth

Multi-Fidelity Simulations

- Computer models have been widely adopted to understand a real-world feature, phenomenon or event.
- Computer simulations are used to solve these models (e.g., finite element / finite difference) .
- The simulation can be either
 - High-fidelity simulation: costly but close to the truth
 - Low-fidelity simulation: cheaper but less accurate

Multi-Fidelity Simulations

- Computer models have been widely adopted to understand a real-world feature, phenomenon or event.
- Computer simulations are used to solve these models (e.g., finite element / finite difference) .
- The simulation can be either
 - High-fidelity simulation: costly but close to the truth
 - Low-fidelity simulation: cheaper but less accurate
 - (intermediate-fidelity simulation)

Motivated Example: Finite Element Simulations

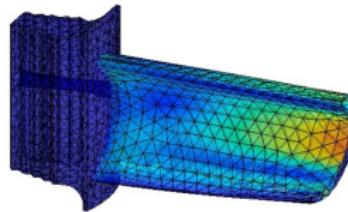
- Thermal stress of jet engine turbine blade can be analyzed through a static structural computer model.
- The model can be *numerically* solved via finite element method.

Motivated Example: Finite Element Simulations

- Thermal stress of jet engine turbine blade can be analyzed through a static structural computer model.
- The model can be *numerically* solved via finite element method.
- **Input:** $\mathbf{x} = (x_1, x_2) = (\text{pressure}, \text{suction})$

Motivated Example: Finite Element Simulations

- Thermal stress of jet engine turbine blade can be analyzed through a static structural computer model.
- The model can be *numerically* solved via finite element method.
- **Input:** $\mathbf{x} = (x_1, x_2) = (\text{pressure}, \text{suction})$
- **Output:** $f(\mathbf{x})$: **maximum** of thermal stress profile
- e.g., $\mathbf{x} = (0.23, 0.71)$

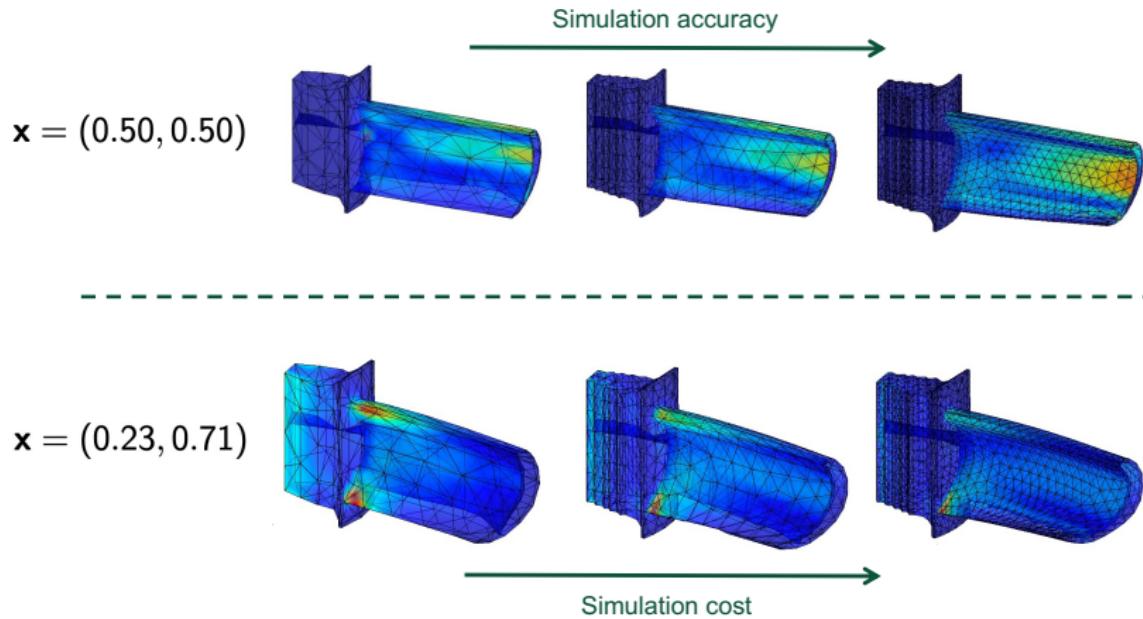


maximum of thermal stress profile $f(0.23, 0.71) = 20.3$

Multi-Fidelity Simulations

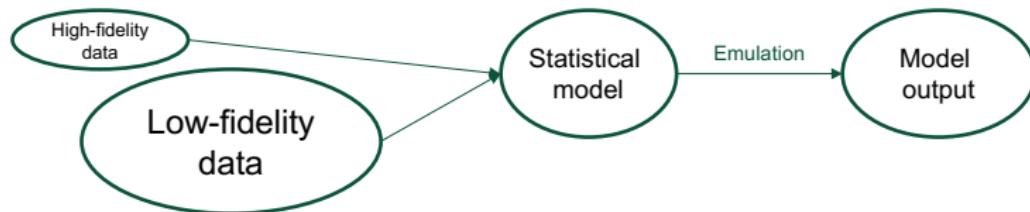
less accurate but cheaper

accurate but expensive



Multi-Fidelity Emulation

- Can we leverage both low- and high-fidelity simulations in order to
 - build a statistical emulator, $\hat{f}(\mathbf{x})$, also known as a **surrogate model**, to approximate the output of a **high-fidelity complex simulator**:
$$\hat{f}(\mathbf{x}) \approx f(\mathbf{x}),$$
where $f(\mathbf{x})$ represents the true simulator, with \mathbf{x} as the input.
- while **minimizing the cost** associated with the simulations?



Existing Methods

- The canonical approach is auto-regressive (AR) model (Kennedy and O'Hagan, 2000).
- AR model assumes additive structure of Gaussian processes (GPs).

$$f_1(\mathbf{x}) = Z_1(\mathbf{x}),$$

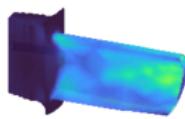
$$f_l(\mathbf{x}) = \rho_{l-1} f_{l-1}(\mathbf{x}) + Z_l(\mathbf{x}), \quad \text{for } 2 \leq l \leq L,$$

where $f_l(\mathbf{x})$ is the scalar simulation output of the computer code at fidelity level l .

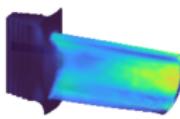
- Several extensions including Qian et al. (2006); Qian and Wu (2008); Le Gratiet (2013); Le Gratiet and Garnier (2014); Perdikaris et al. (2017).

Jet Blade Simulations

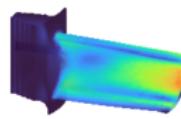
Mesh size = 0.04



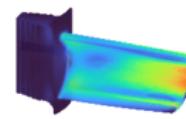
Mesh size = 0.02



Mesh size = 0.01



Mesh size = 0.005



Mesh size = 0

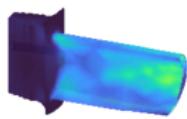


Jet blade simulations with different mesh configurations

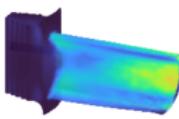
- The fidelity level is often controlled by a tuning parameter.

Jet Blade Simulations

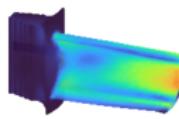
Mesh size = 0.04



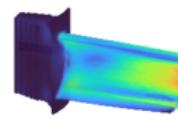
Mesh size = 0.02



Mesh size = 0.01



Mesh size = 0.005



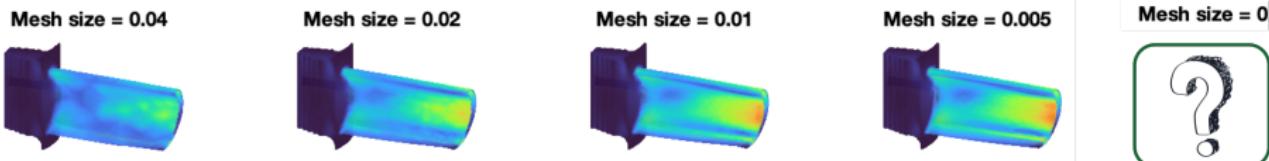
Mesh size = 0



Jet blade simulations with different mesh configurations

- The fidelity level is often controlled by a tuning parameter.
- Q1: Can we account for the tuning parameter and extrapolate the **exact solution** of finite element simulations (i.e., mesh size = 0)?

Jet Blade Simulations



Jet blade simulations with different mesh configurations

- The fidelity level is often controlled by a tuning parameter.
- Q1: Can we account for the tuning parameter and extrapolate the **exact solution** of finite element simulations (i.e., **mesh size = 0**)?
- Q2: How can we choose the design points and tuning parameters when running simulations or computer experiments?



Boutelet, R. and **Sung, C.-L.** (2025)

Active learning for finite element simulations with adaptive
non-stationary kernel function, [arXiv:2503.23158](https://arxiv.org/abs/2503.23158).

MICHIGAN STATE
UNIVERSITY

Non-Stationary Model

Non-Stationary Model (Tuo et al., 2014)

The response variable at input location $\mathbf{x} \in \mathcal{D}$ with mesh size $t \in \mathcal{T}$ is assumed to be:

$$f(\mathbf{x}, t) = \underbrace{\varphi(\mathbf{x})}_{\text{exact solution}} + \underbrace{\delta(\mathbf{x}, t)}_{\text{error}},$$

where $\varphi(\mathbf{x}) := f(\mathbf{x}, 0)$ and $\delta(\mathbf{x}, t)$ are realizations of two mutually independent GPs, respectively.

Remark: Since $f(\mathbf{x}, t)$ must equal to the exact solution $\varphi(\mathbf{x})$ as $t \rightarrow 0$, we need δ to satisfy $\delta(\mathbf{x}, t) \xrightarrow[t \rightarrow 0]{} 0$, for all \mathbf{x} .

Non-Stationary Model (Tuo et al., 2014)

- The **mean function** is assumed to have a separable form, such that

$$\mathbb{E}[\varphi(\mathbf{x})] = g_1(\mathbf{x})^T \boldsymbol{\beta}_1, \quad \mathbb{E}[\delta(\mathbf{x}, t)] = g_2(\mathbf{x}, t)^T \boldsymbol{\beta}_2$$

- The **covariance function** of our response variable is

$$K(\mathbf{x}, \mathbf{x}', t, t') = K_1(\mathbf{x}, \mathbf{x}') + K_2(\mathbf{x}, \mathbf{x}', t, t'),$$

where

Non-Stationary Model (Tuo et al., 2014)

- The **mean function** is assumed to have a separable form, such that

$$\mathbb{E}[\varphi(\mathbf{x})] = g_1(\mathbf{x})^T \boldsymbol{\beta}_1, \quad \mathbb{E}[\delta(\mathbf{x}, t)] = g_2(\mathbf{x}, t)^T \boldsymbol{\beta}_2$$

- The **covariance function** of our response variable is

$$K(\mathbf{x}, \mathbf{x}', t, t') = K_1(\mathbf{x}, \mathbf{x}') + K_2(\mathbf{x}, \mathbf{x}', t, t'),$$

where

- $\varphi(\mathbf{x})$ has a stationary covariance function of the form

$$K_1(\mathbf{x}, \mathbf{x}') = \sigma_1^2 \prod_{i=1}^d e^{-\phi_1^2(x_i - x'_i)^2}$$

- $\delta(\mathbf{x}, t)$ has a **non-stationary covariance function** of the form

$$K_2(\mathbf{x}, \mathbf{x}', t, t') = \sigma_2^2 \mathcal{K}_H(t, t') \prod_{i=1}^d e^{-\phi_2^2(x_i - x'_i)^2}$$

Adaptive Non-Stationary Kernel

- Tuo et al. (2014) proposed a Brownian Motion (BM) kernel,
 $K_{\text{BM}}(t, t') = \min(t, t')^l$.

Adaptive Non-Stationary Kernel

- Tuo et al. (2014) proposed a Brownian Motion (BM) kernel,
 $K_{\text{BM}}(t, t') = \min(t, t')^l$.
- We introduce a new covariance function on the mesh size t , adapted from the Fractional Brownian Motion (FBM):

$$K_H(t, t') = \left\{ \frac{1}{2} (t^{2H} + (t')^{2H} + |t - t'|^{2H}) \right\}^{\frac{l}{2H}}, \quad 0 \leq H \leq 1.$$

The parameter H can be estimated using MLE.

Adaptive Non-Stationary Kernel

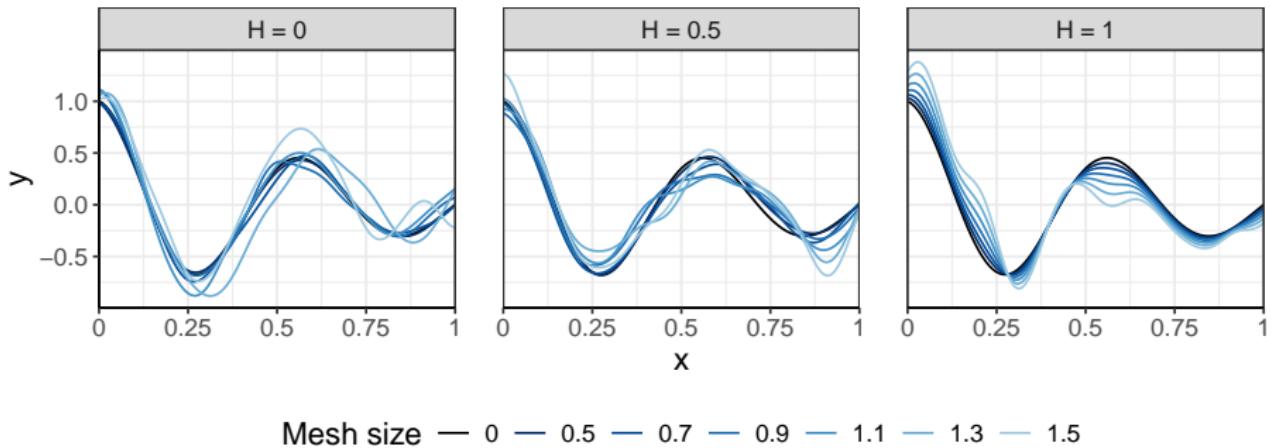
- Tuo et al. (2014) proposed a Brownian Motion (BM) kernel,
 $K_{\text{BM}}(t, t') = \min(t, t')^l$.
- We introduce a new covariance function on the mesh size t , adapted from the Fractional Brownian Motion (FBM):

$$K_H(t, t') = \left\{ \frac{1}{2} (t^{2H} + (t')^{2H} + |t - t'|^{2H}) \right\}^{\frac{l}{2H}}, \quad 0 \leq H \leq 1.$$

The parameter H can be estimated using MLE.

- This BM kernel becomes a special case of FBM kernel ($H = 0.5$).

Adaptive Non-Stationary Kernel



Sample paths of the non-stationary model using the FBM kernel with three distinct values of H .

Prediction

- By the property of conditional normal distributions, the posterior distribution of $f(\mathbf{x}, t)$ given the observations \mathbf{y}_n follows

$$f(\mathbf{x}, t) | \mathbf{y}_n \sim \mathcal{N}(\mu_n(\mathbf{x}, t), \sigma_n^2(\mathbf{x}, t)),$$

$$\mu_n(\mathbf{x}, t) = \mathbf{g}(\mathbf{x}, t)^T \boldsymbol{\beta} + \mathbf{k}_n(\mathbf{x}, t)^T \mathbf{K}_n^{-1} (\mathbf{y}_n - \mathbf{G}_n \boldsymbol{\beta}),$$

and

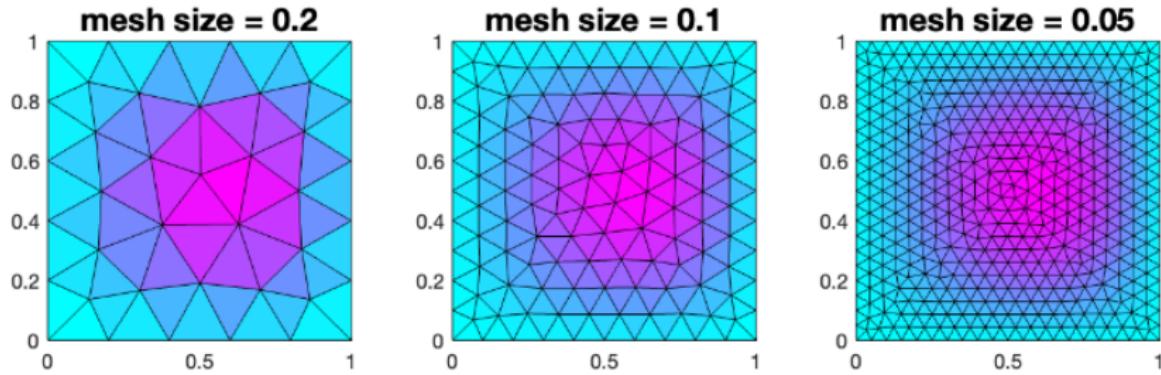
$$\sigma_n^2(\mathbf{x}, t) = K(\mathbf{x}, \mathbf{x}, t, t) - \mathbf{k}_n(\mathbf{x}, t)^T \mathbf{K}_n^{-1} \mathbf{k}_n(\mathbf{x}, t) + \gamma_n(\mathbf{x}, t)^T (\mathbf{G}_n^T \mathbf{K}_n^{-1} \mathbf{G}_n) -$$

- The hyper-parameters in the posterior distribution can be plugged in by their MLE estimates.

Illustration: Poisson Equation

- **Poisson equation:** for $(z_1, z_2) \in D$:

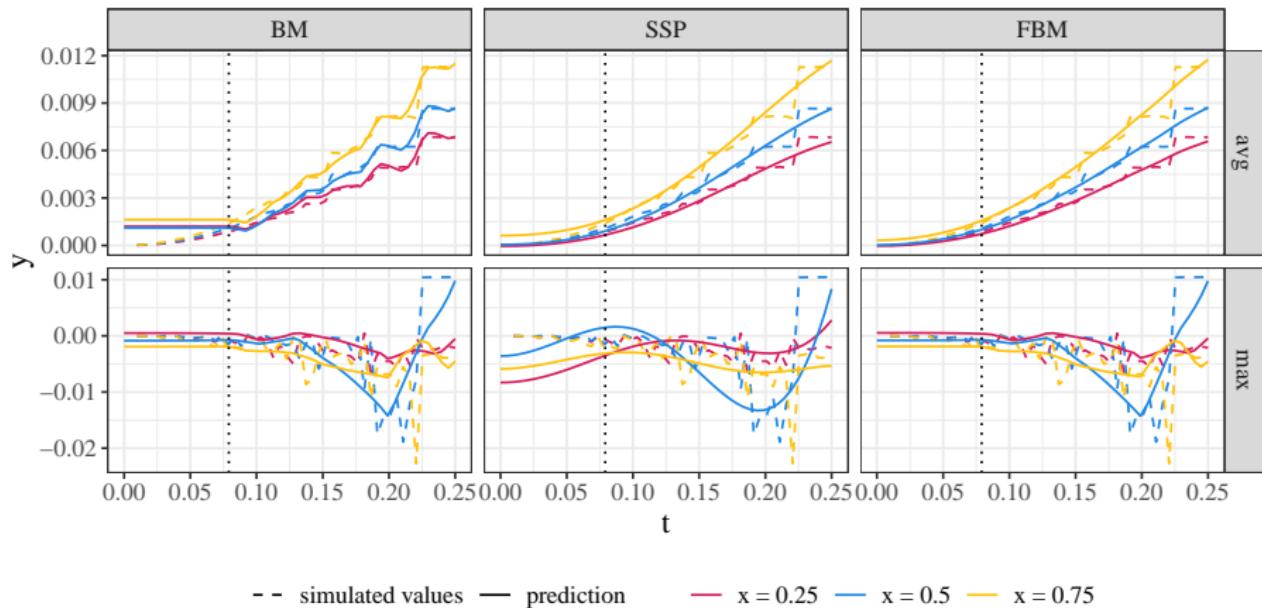
$$\Delta u = (x^2 + 2\pi^2)e^{xz_1} \sin(\pi z_1) \sin(\pi z_2) + 2x\pi e^{xz_1} \cos(\pi z_1) \sin(\pi z_2)$$
$$u = 0 \text{ on } \partial\mathcal{D}$$



The FEM solutions for the three different mesh sizes.

Illustration: Poisson Equation

- Responses of interest $f(\mathbf{x}, t)$ are **average** and **maximum** over D
- The model is trained on mesh sizes $t \in [0.079, 0.25]$
- Predictions are made at $x = 0.25, x = 0.5, x = 0.75$



Active Learning for Finite Element Simulations

- Active learning (also known as Sequential Design)
 - sequentially searches for and acquires new data points at optimal location by a given criterion.
- In multi-fidelity simulation, active learning requires
 - identifying optimal input locations x ,
 - identifying optimal tuning parameters t ,
 - accounting for the respective simulation costs simultaneously.

Active Learning for Finite Element Simulations

- We employ the Integrated Mean Squared Prediction Error (IMSPE) as the foundation of our active learning criterion.
- Specifically, the IMSPE from the n design points, I_n , can be written as

$$I_n := \text{IMSPE}(\mathbf{X}_n, \mathbf{t}_n) = \int_{\mathbf{x} \in \mathcal{D}} \sigma_n^2(\mathbf{x}, 0) d\mathbf{x}.$$

Active Learning for Finite Element Simulations

- We employ the Integrated Mean Squared Prediction Error (IMSPE) as the foundation of our active learning criterion.
- Specifically, the IMSPE from the n design points, I_n , can be written as

$$I_n := \text{IMSPE}(\mathbf{X}_n, \mathbf{t}_n) = \int_{\mathbf{x} \in \mathcal{D}} \sigma_n^2(\mathbf{x}, 0) d\mathbf{x}.$$

- Recall that $\sigma_n^2(\mathbf{x}, 0)$ is the **predictive variance** for the **exact solution** (i.e., $f(\mathbf{x}, t)$ at $t = 0$).

Active Learning for Finite Element Simulations

- Our active learning objective is to find the next best design location $(\mathbf{x}_{n+1}, t_{n+1})$ by minimizing $I_{n+1}(\mathbf{x}_{n+1}, t_{n+1}) := \text{IMSPE}(\mathbf{X}_{n+1}, \mathbf{t}_{n+1})$.

Active Learning for Finite Element Simulations

- Our active learning objective is to find the next best design location $(\mathbf{x}_{n+1}, t_{n+1})$ by minimizing $I_{n+1}(\mathbf{x}_{n+1}, t_{n+1}) := \text{IMSPE}(\mathbf{X}_{n+1}, \mathbf{t}_{n+1})$.

Theorem: IMSPE Reduction

The IMSPE associated with an additional design point $(\tilde{\mathbf{x}}, \tilde{t})$ given the current design $(\mathbf{X}_n, \mathbf{t}_n)$ can be written in an iterative form as (Binois et al., 2019)

$$I_{n+1}(\tilde{\mathbf{x}}, \tilde{t}) = I_n - R_{n+1}(\tilde{\mathbf{x}}, \tilde{t})$$

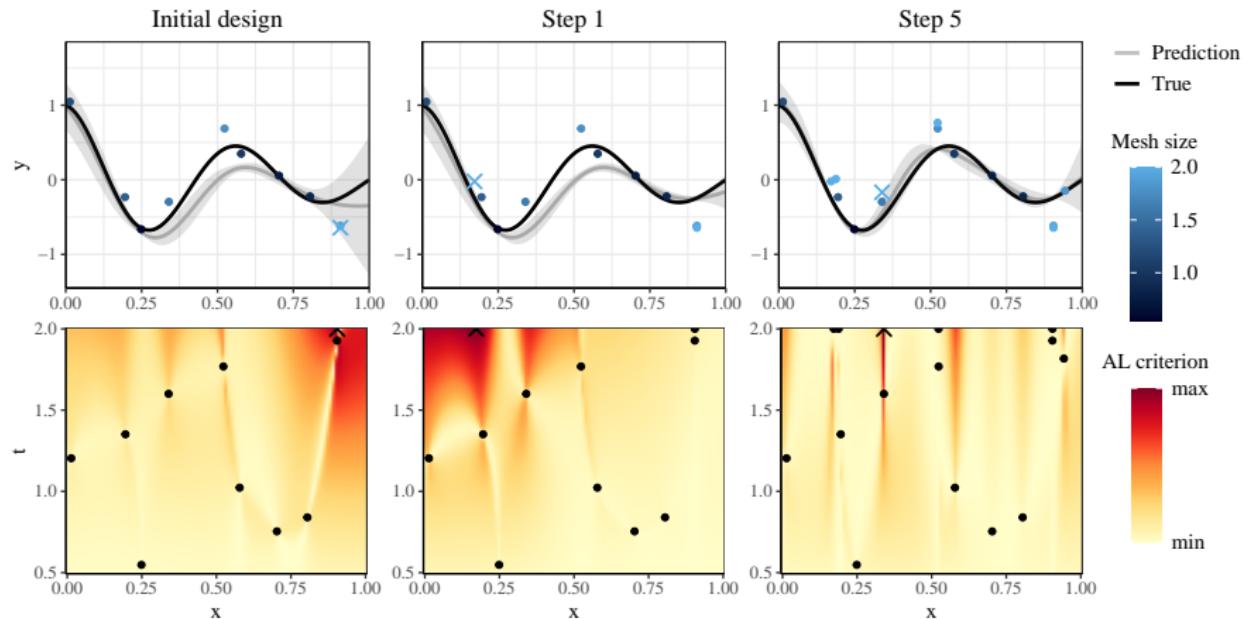
where $R_{n+1}(\tilde{\mathbf{x}}, \tilde{t})$, the **IMSPE reduction**, has a closed-form expression and can be computed with an $\mathcal{O}(n^2)$ cost complexity.

Cost Adjusted IMSPE Reduction

- To take the **computational cost** into account for our criterion, we choose the next point $(\mathbf{x}_{n+1}, t_{n+1})$ by maximizing the ratio between the IMSPE reduction and the cost:

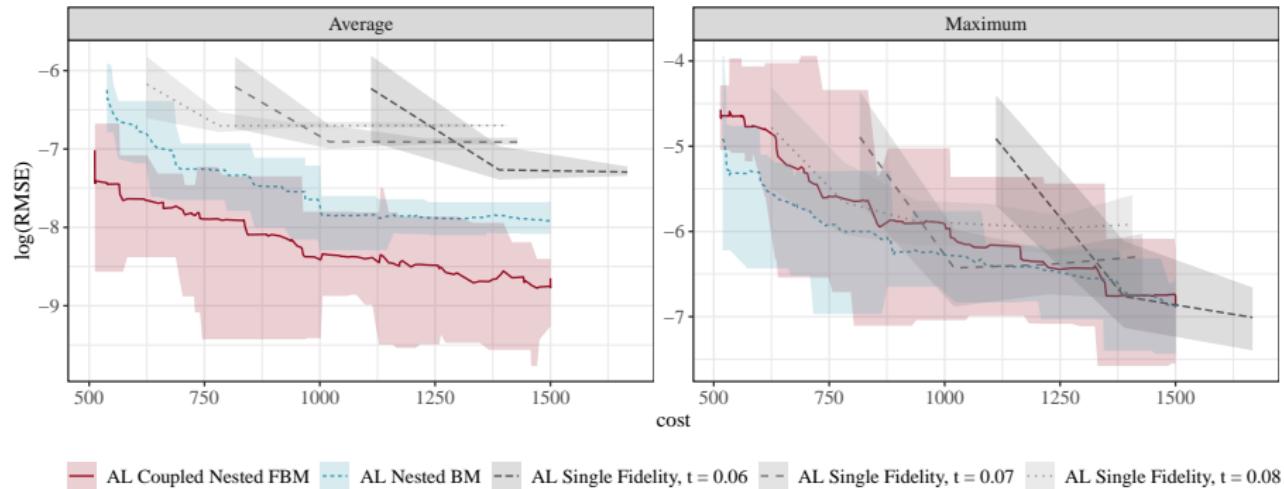
$$(\mathbf{x}_{n+1}, t_{n+1}) = \arg \max_{(\tilde{\mathbf{x}}, \tilde{t}) \in \mathcal{X} \times \mathcal{T}} \frac{R_{n+1}(\tilde{\mathbf{x}}, \tilde{t})}{C(\tilde{t})}.$$

Demonstration

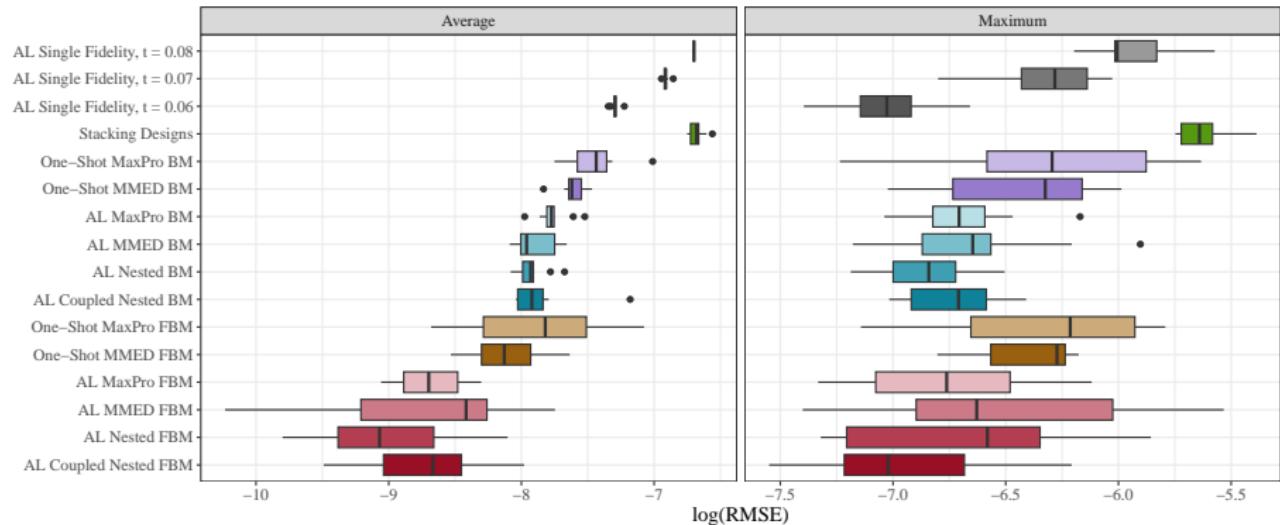


Prediction of our model (top), and active learning criterion surface (bottom). The points represent the current design locations (\bullet), and the best next design point according to the criterion (\times).

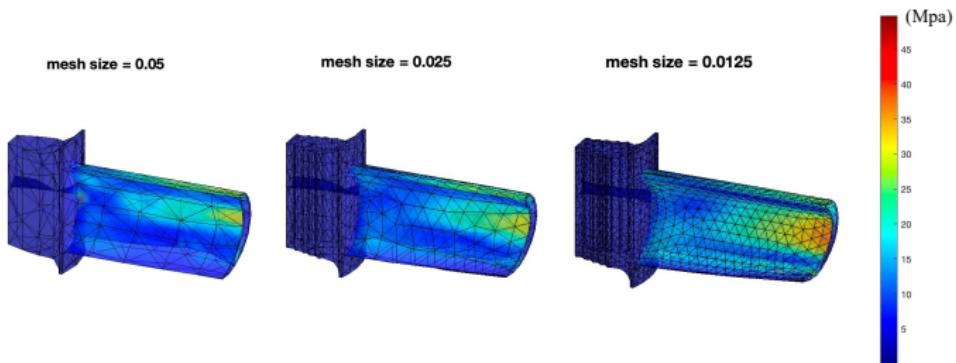
Revisit Poisson Equation



Revisit Poisson Equation

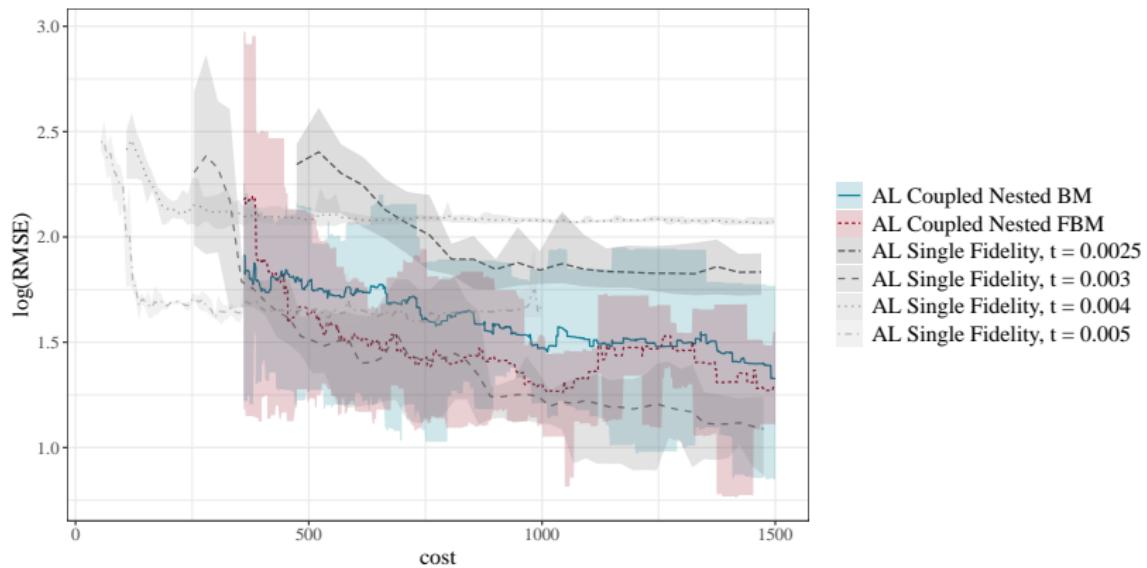


Revisit Motivated Example



- **Input:** $\mathbf{x} = (x_1, x_2) = (\text{pressure}, \text{suction}) \in \Omega = [0.25, 0.75]^2$
- **Output:** $f(\mathbf{x})$: **maximum** of the thermal stress profile
- **Test data:** Simulations with mesh size $t = 0.001$ at 50 uniform test input locations are conducted to examine the performance

Revisit Motivated Example



RMSE (in logarithmic scale) for the jet engine turbine blade case study with respect to the simulation cost. Solid lines indicate the average over 5 repetitions, while shaded regions represent the range.

Discussion on Non-Stationary GP Model

- Recall the model:

$$f(\mathbf{x}, \textcolor{blue}{t}) = \underbrace{\varphi(\mathbf{x})}_{\text{exact solution}} + \underbrace{\delta(\mathbf{x}, \textcolor{blue}{t})}_{\text{error}}.$$

Discussion on Non-Stationary GP Model

- Recall the model:

$$f(\mathbf{x}, \textcolor{blue}{t}) = \underbrace{\varphi(\mathbf{x})}_{\text{exact solution}} + \underbrace{\delta(\mathbf{x}, \textcolor{blue}{t})}_{\text{error}}.$$

- **Q:** Can we consider a flexible, nonadditive model?



Heo, J., Boutelet, R. and **Sung, C.-L.** (2025)

Diffusion non-additive model for multi-fidelity computer experiments
with tuning parameters, **in preparation**.



Diffusion Non-Additive (DNA) Model

Diffusion Non-Additive (DNA) Model

The response variable at input location $\mathbf{x} \in \mathcal{D}$ with mesh size $t_l \in \mathcal{T}$ is assumed to be:

$$f(\mathbf{x}, t_l) = W(t_l, \mathbf{x}, f(\mathbf{x}, t_{l-1})), \quad l = 2, \dots, L.$$

where $t_1 > t_2 > \dots > t_L > 0$ and W is a realization of a GP.

Remark 1: The relationships between fidelity levels are no longer additive.

Remark 2: The term “diffusion” is inspired by its analogy in generative models.

Nonseparable Kernel

- The **mean function** of W is assumed to a constant α .
- The **covariance function** of W is assumed to be a **nonseparable kernel** function (Cressie and Huang, 1999; Gneiting, 2002):

$$K((t, \mathbf{x}, y), (t', \mathbf{x}', y')) = \left(\frac{(t - t')^2}{\theta_t} + 1 \right)^{-\left(\frac{\beta(d+1)}{2} + \delta\right)} \times \\ \exp \left(- \left(\frac{(t - t')^2}{\theta_t} + 1 \right)^{-\beta} \left[\frac{(y - y')^2}{\theta_y} + \sum_{j=1}^d \frac{(x_j - x'_j)^2}{\theta_j} \right] \right).$$

Nonseparable Kernel

- The **mean function** of W is assumed to a constant α .
- The **covariance function** of W is assumed to be a **nonseparable kernel** function (Cressie and Huang, 1999; Gneiting, 2002):

$$K((t, \mathbf{x}, y), (t', \mathbf{x}', y')) = \left(\frac{(t - t')^2}{\theta_t} + 1 \right)^{-\left(\frac{\beta(d+1)}{2} + \delta\right)} \times \\ \exp \left(- \left(\frac{(t - t')^2}{\theta_t} + 1 \right)^{-\beta} \left[\frac{(y - y')^2}{\theta_y} + \sum_{j=1}^d \frac{(x_j - x'_j)^2}{\theta_j} \right] \right).$$

- $0 \leq \beta \leq 1$ allows for the interaction of (\mathbf{x}, y) and the tuning parameter t .
- When $\beta = 0$, it becomes separable.
- Parameters can be estimated via MLE.

Posterior of $f(\mathbf{x}, t)$

- Let $f_l(\mathbf{x}) := f(\mathbf{x}, t_l)$.
- Based on the properties of conditional multivariate normal distribution, it follows that

$$f_l(\mathbf{x}) | Y_N, \mathbf{f}_{l-1}(\mathbf{x}) \sim \mathcal{N}(\mu_l(\mathbf{x}, \mathbf{f}_{l-1}(\mathbf{x})), \sigma_l^2(\mathbf{x}, \mathbf{f}_{l-1}(\mathbf{x})))$$

for $l = 2, \dots, L$ and

$$f(\mathbf{x}, t) | Y_N, \mathbf{f}_L(\mathbf{x}) \sim \mathcal{N}(\mu_{L+1}(t, \mathbf{x}, \mathbf{f}_L(\mathbf{x})), \sigma_{L+1}^2(t, \mathbf{x}, \mathbf{f}_L(\mathbf{x}))).$$

Posterior of $f(\mathbf{x}, t)$

- Let $f_l(\mathbf{x}) := f(\mathbf{x}, t_l)$.
- Based on the properties of conditional multivariate normal distribution, it follows that

$$f_l(\mathbf{x}) | Y_N, \mathbf{f}_{l-1}(\mathbf{x}) \sim \mathcal{N}(\mu_l(\mathbf{x}, \mathbf{f}_{l-1}(\mathbf{x})), \sigma_l^2(\mathbf{x}, \mathbf{f}_{l-1}(\mathbf{x})))$$

for $l = 2, \dots, L$ and

$$f(\mathbf{x}, t) | Y_N, \mathbf{f}_L(\mathbf{x}) \sim \mathcal{N}(\mu_{L+1}(t, \mathbf{x}, \mathbf{f}_L(\mathbf{x})), \sigma_{L+1}^2(t, \mathbf{x}, \mathbf{f}_L(\mathbf{x}))).$$

- Intractable posterior distribution $p(f(\mathbf{x}, t) | Y_N)$ can be approximated by Monte Carlo integration.
- However, it can be **computationally demanding!**

The closed form expression of DNA model

Proposition 1: The closed-form expressions

- Suppose that the design points are nested. The posterior mean and variance of $f(\mathbf{x}, t)$ given the data Y_N for $0 \leq t < t_L$ can be expressed in a recursive fashion:

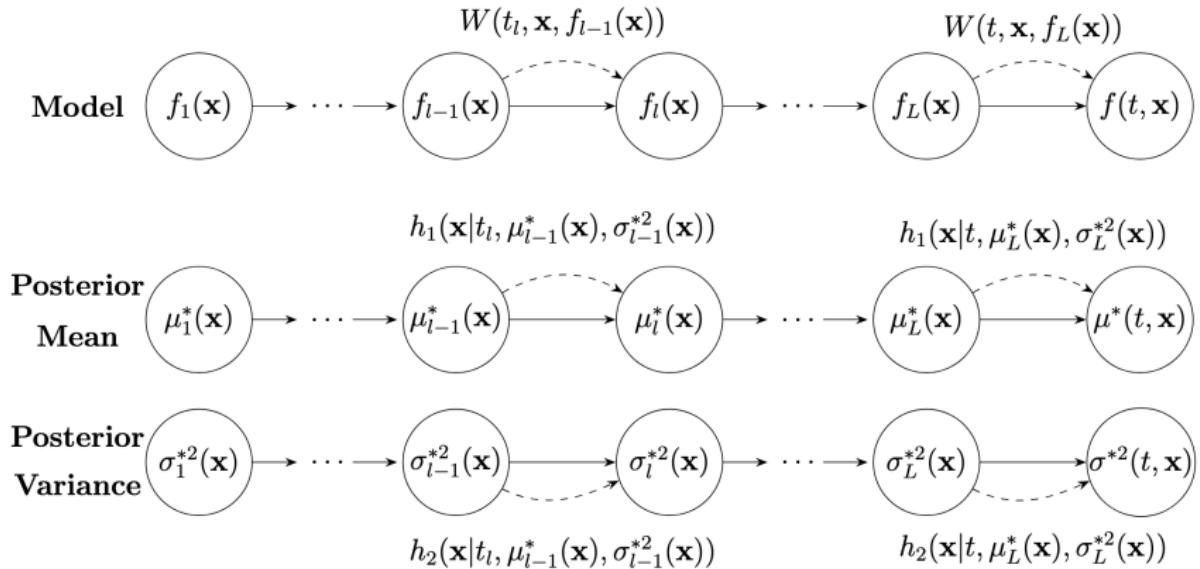
$$\begin{aligned}\mu^*(t, \mathbf{x}) &= \mathbb{E}[f(\mathbf{x}, t) | Y_N] = h_1(\mathbf{x} | t, \mu_L^*(\mathbf{x}), \sigma_L^{*2}(\mathbf{x})), \\ \sigma^{*2}(t, \mathbf{x}) &= \mathbb{V}[f(\mathbf{x}, t) | Y_N] = h_2(\mathbf{x} | t, \mu_L^*(\mathbf{x}), \sigma_L^{*2}(\mathbf{x})),\end{aligned}$$

and for $l = 2, \dots, L$,

$$\begin{aligned}\mu_l^*(\mathbf{x}) &= \mathbb{E}[f_l(\mathbf{x}) | Y_N] = h_1(\mathbf{x} | t_l, \mu_{l-1}^*(\mathbf{x}), \sigma_{l-1}^{*2}(\mathbf{x})), \\ \sigma_l^{*2}(\mathbf{x}) &= \mathbb{V}[f_l(\mathbf{x}) | Y_N] = h_2(\mathbf{x} | t_l, \mu_{l-1}^*(\mathbf{x}), \sigma_{l-1}^{*2}(\mathbf{x})),\end{aligned}$$

where $h_1(\mathbf{x} | t, \mu, \sigma^2)$ and $h_2(\mathbf{x} | t, \mu, \sigma^2)$ both have closed-form expressions.

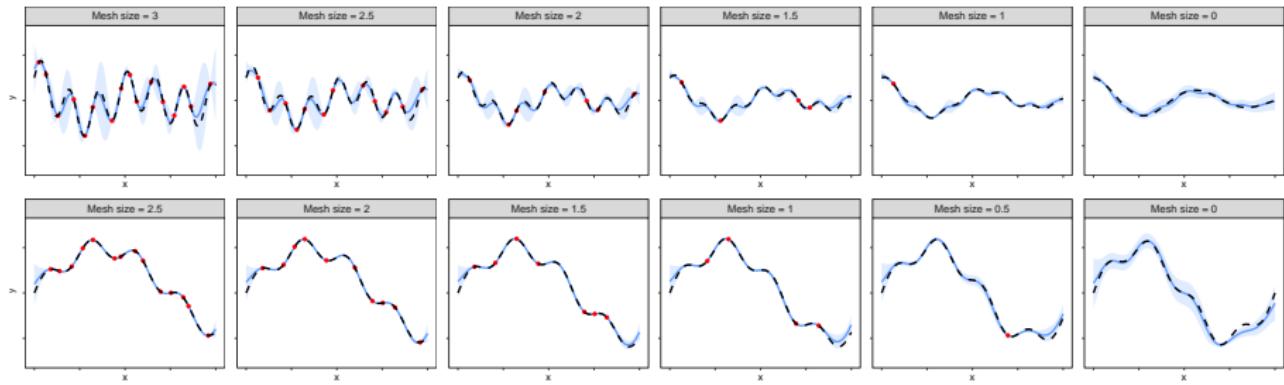
Diffusion Representation



The Markov chain representation of the hierarchical modeling structure.

Synthetic Examples

$$\begin{cases} f(x, t) = \exp(-1.4x) \cos(3.5\pi x) + t^2 \sin(40x)/10, & x \in [0, 1], \\ f(x, t) = \sin\left(\frac{10\pi x}{5+t}\right) + 0.2 \sin(8\pi x), & x \in [0, 1]. \end{cases}$$



Conclusion

- We introduce a new, adaptive non-stationary kernel function (**FBM kernel**) and the IMSPE-based active learning for multi-fidelity simulations with **continuous** fidelity levels.
- We propose a new, flexible model (**DNA model**), which provides the **closed-form expressions** for both the posterior mean and variance under a nonseparable kernel function.
- For cases where the design is not nested, the stochastic EM algorithm is employed.
- R packages will be available for both works.



Thank You!

Thank NSF DMS 2338018 for supporting this work.



MICHIGAN STATE
UNIVERSITY

- Binois, M., Huang, J., Gramacy, R. B., and Ludkovski, M. (2019). Replication or exploration? sequential design for stochastic simulation experiments. *Technometrics*, 61(1):7–23.
- Cressie, N. and Huang, H.-C. (1999). Classes of nonseparable, spatio-temporal stationary covariance functions. *Journal of the American Statistical Association*, 94(448):1330–1339.
- Gneiting, T. (2002). Nonseparable, stationary covariance functions for space–time data. *Journal of the American Statistical Association*, 97(458):590–600.
- Kennedy, M. C. and O'Hagan, A. (2000). Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13.
- Le Gratiet, L. (2013). Bayesian analysis of hierarchical multifidelity codes. *SIAM/ASA Journal on Uncertainty Quantification*, 1(1):244–269.
- Le Gratiet, L. and Garnier, J. (2014). Recursive co-kriging model for design of computer experiments with multiple levels of fidelity. *International Journal for Uncertainty Quantification*, 4(5):365–386.

- Perdikaris, P., Raissi, M., Damianou, A., Lawrence, N. D., and Karniadakis, G. E. (2017). Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2198):20160751.
- Qian, P. Z. G. and Wu, C. F. J. (2008). Bayesian hierarchical modeling for integrating low-accuracy and high-accuracy experiments. *Technometrics*, 50(2):192–204.
- Qian, Z., Seepersad, C. C., Joseph, V. R., Allen, J. K., and Wu, C. F. J. (2006). Building surrogate models based on detailed and approximate simulations. *Journal of Mechanical Design*, 128(4):668–677.
- Tuo, R., Wu, C. F. J., and Yu, D. (2014). Surrogate modeling of computer experiments with different mesh densities. *Technometrics*, 56(3):372–380.