Advancing Multi-Fidelity Computer Experiments: Applications to Uncertainty Quantification

Chih-Li Sung

Department of Statistics and Probability
Michigan State University

U. of South Carolina, Department of Statistics, January 23, 2025

Outline

Digital Twins

- The term digital twins has been getting much attention in engineering and manufacturing for a few years as companies realize the potential of virtually replicating a real-world environment.
- The global market for digital twins in industry alone is projected to grow to $156 billion by 2030.

NSF 24-559: Mathematical Foundations of Digital Twins

Program Solicitation

Document Information

Document History
- **Posted:** March 22, 2024

Download the solicitation (PDF, 0.8mb) | View the program page

National Science Foundation
Directorate for Mathematical and Physical Sciences
Division of Mathematical Sciences
Directorate for Engineering
Division of Civil, Mechanical and Manufacturing Innovation

Air Force Office of Scientific Research

What Are Digital Twins

- A digital twin is a real-time virtual representation of a physical object or system.
- It simulates and monitors real-world processes, enabling control, testing, and optimization without physical risks.

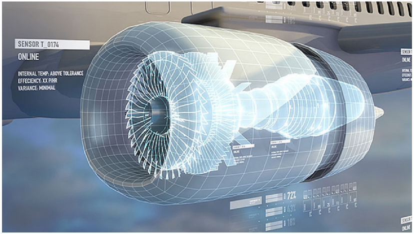
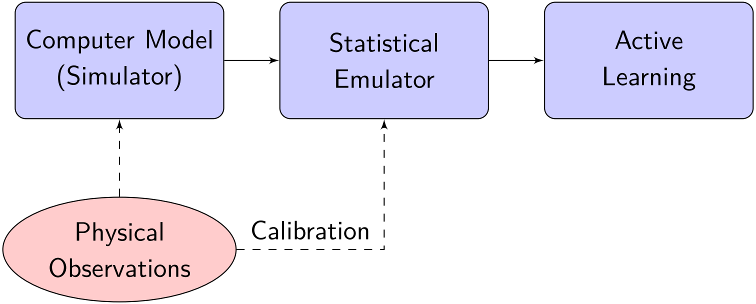


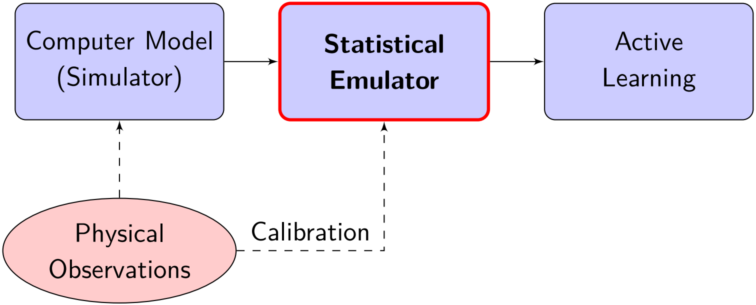
Figure: *Digital twin of an aircraft engine used to monitor performance and troubleshoot issues in real time. Credit: GE.*

Uncertainty Quantification (UQ)

- Uncertainty Quantification (UQ) is a critical component that powers the accuracy and reliability of digital twins.



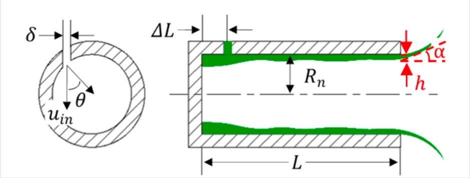
Statistical Emulator/ Surrogate Model



Rocket Injector Simulator

- We consider here a simplex swirl injector system for liquid-propellant rocket engines.[1]

Swirl Injector Configuration



L	20-100 mm
R_n	2-5 mm
θ	45°-75°
δ	0.5-2 mm
ΔL	1-4 mm

[1]Mak, **Sung**, et al. (2018). An efficient surrogate model for emulation and physics extraction of large eddy simulations. *JASA*.

Rocket Injector Simulator

- High-fidelity flow simulations are conducted using the theoretical and numerical framework for modeling high-pressure mixing and combustion processes.

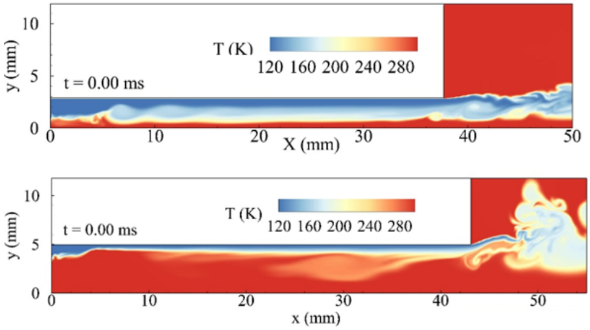


Figure: *Temperature snapshots for two design settings.*

Rocket Injector Simulator

- A key challenge here is that the high-fidelity flow simulations are too time-consuming for design purposes.
- Each simulation requires 28,800 CPU hours to obtain 1,000 snapshots with time-interval 0.03ms.

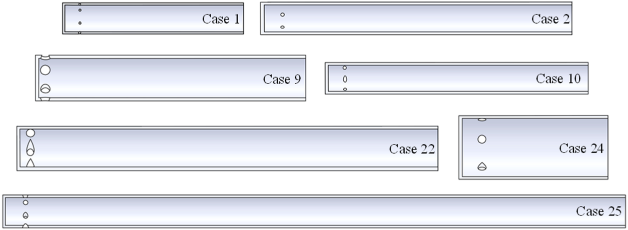
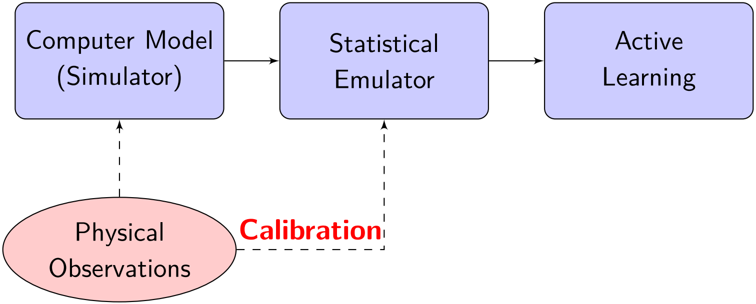


Figure: *Computational domain with different design variables.*

Statistical Emulator/ Surrogate Model

- A statistical emulator, also known as a surrogate model, is constructed to approximate the output of a complex simulator.
- The goal is to "emulate" the true simulator, including the uncertainty in the approximation:
$$\hat{f}(\mathbf{x}) \approx f(\mathbf{x}),$$
where $f(\mathbf{x})$ represents the true simulator, with $\mathbf{x}$ as the input (e.g., design variables), and $\hat{f}(\mathbf{x})$ is the emulator/surrogate model.
- Gaussian Processes (GPs) are widely used for building such emulators due to their flexibility and ability to quantify uncertainty.

Model Calibration



Model Calibration

- Computer models are useful for simulating complex systems, but how do we ensure they accurately represent reality?
- **Calibration** aligns the computer model's output with real-world observations, making it more reliable.

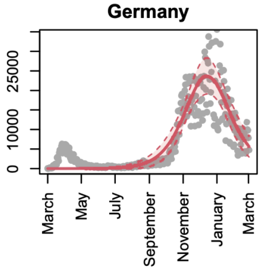
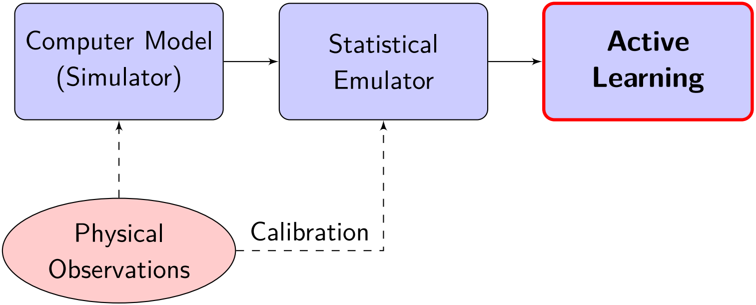


Figure: *Calibration of COVID-19 model.*[2]

[2] **Sung** and Hung (2024). Efficient calibration for imperfect epidemic models with applications to the analysis of COVID-19. *JRSSC*.

Active Learning

- How can we enhance the accuracy of the statistical emulator?
- By strategically selecting "informative" samples $\mathbf{x}_i$, we can improve the emulator's performance, $\hat{f}(\mathbf{x})$.



- In the following, I will present two recent works that demonstrate advancements in active learning techniques.

Sung, C.-L., Ji, Y., Mak, S., Wang, W., and Tang, T. (2024)

Stacking designs: designing multifidelity computer experiments with target predictive accuracy, *JUQ*, 12(1), 157-181.

MICHIGAN STATE UNIVERSITY

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**: $f(\mathbf{x})$: average of thermal stress
- e.g., $\mathbf{x} = (0.23, 0.71)$

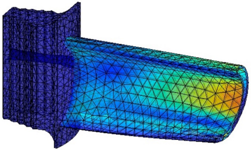


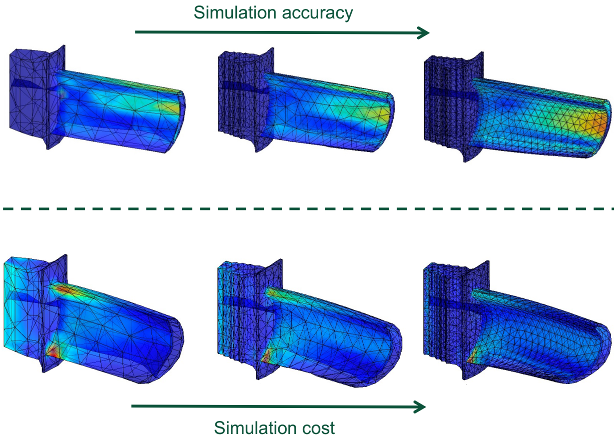
Figure: *average of thermal stress* $f(0.23, 0.71) = 10.5$

Multi-Fidelity Simulations via Mesh Configuration

less accurate but cheaper accurate but expensive

$\mathbf{x} = (0.50, 0.50)$

$\mathbf{x} = (0.23, 0.71)$

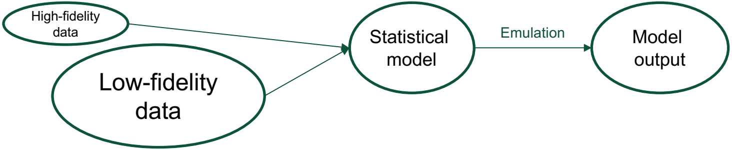


Statistical Emulation

- Can we leverage both low- and high-fidelity simulations in order to

Statistical Emulation

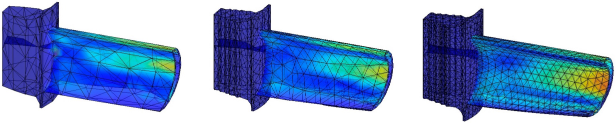
- Can we leverage both low- and high-fidelity simulations in order to
  - maximize the accuracy of model predictions,
  - while minimizing the cost associated with the simulations?



Notation

fidelity level	1		2		3
output	$f_1(\mathbf{x})$		$f_2(\mathbf{x})$		$f_3(\mathbf{x})$
mesh size	h_1	$>$	h_2	$>$	h_3
cost	C_1	$<$	C_2	$<$	C_3

$\mathbf{x} = (0.50, 0.50)$



Existing Methods

- Modeling:
  - Co-kriging or Auto-Regressive (AR) model[3]:

    $$f_l(\mathbf{x}) = \rho_{l-1} f_{l-1}(\mathbf{x}) + Z_{l-1}(\mathbf{x}), \quad l = 2, \ldots, L$$

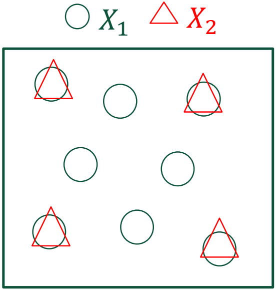
    where both $f_{l-1}(\mathbf{x})$ and $Z_{l-1}(\mathbf{x})$ follow Gaussian Process (GP) priors.
  - GP priors are commonly used in the Bayesian framework to model unknown functions.
  - The posterior of the auto-regressive model is a **normal distribution** with closed-form posterior mean and variance.

[3]Kennedy and O'Hagan (2000). Predicting the output from a complex computer code when fast approximations are available. *Biometrika*

Existing Methods

- Experimental Design (Nested Space-Filling Design):

$$X_L \subseteq X_{L-1} \subseteq \cdots \subseteq X_1$$



- This design improves computational efficiency because it allows us to compute the difference between any two levels (i.e., $f_l(X_l) - f_{l-1}(X_l)$).

Questions

- Q1: How to emulate the exact solution, i.e., $f_\infty(\mathbf{x})$ when $h_\infty \to 0$?[4]

[4] Tuo, Wu, and Yu (2014). Surrogate modeling of computer experiments with different mesh densities. *Technometrics*

Questions

- Q1: How to emulate the exact solution, i.e., $f_\infty(\mathbf{x})$ when $h_\infty \to 0$?[4]
- Q2: Sample size of each level?

[4] Tuo, Wu, and Yu (2014). Surrogate modeling of computer experiments with different mesh densities. *Technometrics*

Questions

- Q1: How to emulate the exact solution, i.e., $f_\infty(\mathbf{x})$ when $h_\infty \to 0$?[4]
- Q2: Sample size of each level?
- Q3: How many fidelity levels?

[4] Tuo, Wu, and Yu (2014). Surrogate modeling of computer experiments with different mesh densities. *Technometrics*

Questions

- Q1: How to emulate the exact solution, i.e., $f_\infty(\mathbf{x})$ when $h_\infty \to 0$?[4]
- Q2: Sample size of each level?
- Q3: How many fidelity levels?
- Q4: Mesh size/density specification?

[4] Tuo, Wu, and Yu (2014). Surrogate modeling of computer experiments with different mesh densities. *Technometrics*

Questions

- Q1: How to emulate the exact solution, i.e., $f_\infty(\mathbf{x})$ when $h_\infty \to 0$?[4]
- Q2: Sample size of each level?
- Q3: How many fidelity levels?
- Q4: Mesh size/density specification?
- Q5: Is it better than single-fidelity simulation?

[4] Tuo, Wu, and Yu (2014). Surrogate modeling of computer experiments with different mesh densities. *Technometrics*

Multi-Level Interpolator

- Emulator: Multi-Level (ML) interpolator
- Goal: Emulate $f_\infty(\mathbf{x})$

Multi-Level Interpolator

- Emulator: Multi-Level (ML) interpolator
- Goal: Emulate $f_\infty(\mathbf{x})$
- Idea: with $f_0(\mathbf{x}) = 0$

$$
\begin{aligned}
f_L(\mathbf{x}) =& (f_1(\mathbf{x}) - f_0(\mathbf{x})) + (f_2(\mathbf{x}) - f_1(\mathbf{x})) + \cdots + (f_L(\mathbf{x}) - f_{L-1}(\mathbf{x})) \\
:=& Z_1(\mathbf{x}) + Z_2(\mathbf{x}) + \cdots + Z_L(\mathbf{x}),
\end{aligned}
$$

where $Z_l(\mathbf{x}) := (f_l(\mathbf{x}) - f_{l-1}(\mathbf{x}))$.

- Assume the data is nested $X_L \subseteq X_{L-1} \subseteq \cdots \subseteq X_1$

Multi-Level Interpolator

- Emulator: Multi-Level (ML) interpolator
- Goal: Emulate $f_\infty(\mathbf{x})$
- Idea: with $f_0(\mathbf{x}) = 0$

$$\begin{aligned} f_L(\mathbf{x}) =& (f_1(\mathbf{x}) - f_0(\mathbf{x})) + (f_2(\mathbf{x}) - f_1(\mathbf{x})) + \cdots + (f_L(\mathbf{x}) - f_{L-1}(\mathbf{x})) \\ :=& Z_1(\mathbf{x}) + Z_2(\mathbf{x}) + \cdots + Z_L(\mathbf{x}), \end{aligned}$$

  where $Z_l(\mathbf{x}) := (f_l(\mathbf{x}) - f_{l-1}(\mathbf{x}))$.
- Assume the data is nested $X_L \subseteq X_{L-1} \subseteq \cdots \subseteq X_1$
- $Z_l(\mathbf{x})$ is observed at the nested sites X_l, that is,

$$Z_l(X_l) = f_l(X_l) - f_{l-1}(X_l).$$

Multi-Level Interpolator

- The reproducing kernel Hilbert space (RKHS) interpolator for each $Z_l(\mathbf{x}) = (f_l(\mathbf{x}) - f_{l-1}(\mathbf{x}))$ is

$$\hat{Z}_l(\mathbf{x}) = \Phi_l(\mathbf{x}, X_l)\Phi_l(X_l, X_l)^{-1} Z(X_l),$$

  where Φ_l is a positive definite kernel function.

Multi-Level Interpolator

- The reproducing kernel Hilbert space (RKHS) interpolator for each $Z_l(\mathbf{x}) = (f_l(\mathbf{x}) - f_{l-1}(\mathbf{x}))$ is

$$\hat{Z}_l(\mathbf{x}) = \Phi_l(\mathbf{x}, X_l)\Phi_l(X_l, X_l)^{-1}Z(X_l),$$

  where Φ_l is a positive definite kernel function.

- ML Interpolator:

$$\hat{f}_L(\mathbf{x}) = \hat{Z}_1(\mathbf{x}) + \hat{Z}_2(\mathbf{x}) + \cdots + \hat{Z}_L(\mathbf{x}).$$

Matérn kernel

Assumption: Matérn kernel Φ

$$\Phi_l(\mathbf{x},\mathbf{x}') = \phi_l(\|\theta_l \odot (\mathbf{x}-\mathbf{x}')\|_2)$$

with

$$\phi_l(r) = \frac{\sigma_l^2}{\Gamma(\nu_l)2^{\nu_l-1}}(2\sqrt{\nu_l}r)_l^{\nu}B_{\nu_l}(2\sqrt{\nu_l}r),$$

- ν_l: smoothness parameter
- θ_l: lengthscale parameter
- σ_l^2: scalar parameter
- B_ν: the modified Bessel function of the second kind
- Parameters can be estimated via either CV or MLE (by a GP assumption)

Error Analysis of ML Interpolator

- ML Interpolator $\hat{f}_L(\mathbf{x}) = \hat{Z}_1(\mathbf{x}) + \hat{Z}_2(\mathbf{x}) + \cdots + \hat{Z}_L(\mathbf{x})$
- Recall our goal is to emulate $f_\infty(\mathbf{x})$

Error Analysis of ML Interpolator

- ML Interpolator $\hat{f}_L(\mathbf{x}) = \hat{Z}_1(\mathbf{x}) + \hat{Z}_2(\mathbf{x}) + \cdots + \hat{Z}_L(\mathbf{x})$
- Recall our goal is to emulate $f_\infty(\mathbf{x})$

$$|f_\infty(\mathbf{x}) - \hat{f}_L(\mathbf{x})| \leq \underbrace{|f_\infty(\mathbf{x}) - f_L(\mathbf{x})|}_{\text{simulation error}} + \underbrace{|f_L(\mathbf{x}) - \hat{f}_L(\mathbf{x})|}_{\text{emulation error}}.$$

Error Analysis of ML Interpolator

- ML Interpolator $\hat{f}_L(\mathbf{x}) = \hat{Z}_1(\mathbf{x}) + \hat{Z}_2(\mathbf{x}) + \cdots + \hat{Z}_L(\mathbf{x})$
- Recall our goal is to emulate $f_\infty(\mathbf{x})$

$$|f_\infty(\mathbf{x}) - \hat{f}_L(\mathbf{x})| \leq \underbrace{|f_\infty(\mathbf{x}) - f_L(\mathbf{x})|}_{\text{simulation error}} + \underbrace{|f_L(\mathbf{x}) - \hat{f}_L(\mathbf{x})|}_{\text{emulation error}}.$$

- (analogue to statistical learning)

Idea of Stacking Design

- Given a desired accuracy $\epsilon > 0$

Idea of Stacking Design

- Given a desired accuracy $\epsilon > 0$
- We wish $\|f_\infty - \hat{f}_L\| < \epsilon$ (i.e., with target predictive accuracy!)

Idea of Stacking Design

- Given a desired accuracy $\epsilon > 0$
- We wish $\|f_\infty - \hat{f}_L\| < \epsilon$ (i.e., with target predictive accuracy!)

$$\|f_\infty - f_L\| < \frac{\epsilon}{2}$$

determine L

$$\|f_L - \hat{f}_L\| < \frac{\epsilon}{2}$$

determine sample sizes n_l

Stacking Design

Control emulation error $\|f_L - \hat{f}_L\|$

Proposition 1: Emulation error

Suppose that

- the input space is d-dimensional and is bounded and convex,
- X_l is **quasi-uniform** with sample size n_l,

Then,

$$|f_L(\mathbf{x}) - \hat{f}_L(\mathbf{x})| \leq c \sum_{l=1}^{L} \|\theta_l\|_2^{\nu_l} n_l^{-\nu_l/d} \|f_l - f_{l-1}\|_{\mathcal{N}_{\Phi_l}(\Omega)},$$

where $\|\cdot\|\|_{\mathcal{N}_{\Phi_l}(\Omega)}$ is the RKHS norm.

Sample size determination n_l

- Sample size n_l can be determined by minimizing the error bound and the total cost by the method of Lagrange multipliers

$$\sum_{l=1}^{L}\|\theta_l\|_2^{\nu_{\min}} n_l^{-\nu_{\min}/d}\|f_l - f_{l-1}\|_{\mathcal{N}_{\Phi_l}(\Omega)} + \lambda\sum_{l=1}^{L} n_l C_l,$$

where $\nu_{\min} = \min_{l=1,\ldots,L}\nu_l$, which gives

Sample size determination n_l

- Sample size n_l can be determined by minimizing the error bound and the total cost by the method of Lagrange multipliers

$$\sum_{l=1}^{L} \|\theta_l\|_2^{\nu_{\min}} n_l^{-\nu_{\min}/d} \|f_l - f_{l-1}\|_{\mathcal{N}_{\Phi_l}(\Omega)} + \lambda \sum_{l=1}^{L} n_l C_l,$$

where $\nu_{\min} = \min_{l=1,\ldots,L} \nu_l$, which gives

$$n_l = \mu \left(\frac{\|\theta_l\|^{\nu_{\min}}}{C_l} \|f_l - f_{l-1}\|_{\mathcal{N}_{\Phi_l}(\Omega)} \right)^{d/(\nu_{\min}+d)}$$

for some constant $\mu > 0$.

Sample size determination n_l

- Sample size n_l can be determined by minimizing the error bound and the total cost by the method of Lagrange multipliers

$$\sum_{l=1}^{L} \|\theta_l\|_2^{\nu_{\min}} n_l^{-\nu_{\min}/d} \|f_l - f_{l-1}\|_{\mathcal{N}_{\Phi_l}(\Omega)} + \lambda \sum_{l=1}^{L} n_l C_l,$$

where $\nu_{\min} = \min_{l=1,\ldots,L} \nu_l$, which gives

$$n_l = \mu \left(\frac{\|\theta_l\|^{\nu_{\min}}}{C_l} \|f_l - f_{l-1}\|_{\mathcal{N}_{\Phi_l}(\Omega)} \right)^{d/(\nu_{\min}+d)}$$

for some constant $\mu > 0$.

- Find μ such that $\|f_L - \hat{f}_L\| < \epsilon/2$

Questions

- Q1: How to emulate the exact solution, i.e., $f_\infty(\mathbf{x})$? $\hat{f}_L$
- Q2: Sample size of each level? n_l
- Q3: How many fidelity levels?
- Q4: Mesh size/density specification?
- Q5: Is it better than single-fidelity simulation?

Control Simulation Error $\|f_\infty - f_L\|$

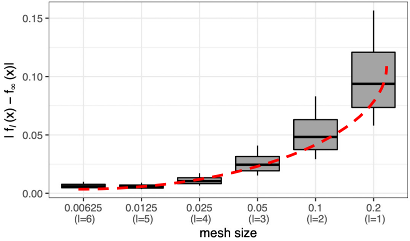
Error Rate of Finite Element Simulations

Under some regularity conditions, for a constant $\alpha \in \mathbb{N}$,[a]

$$|f_\infty(\mathbf{x}) - f_L(\mathbf{x})| < c(\mathbf{x})h_L^\alpha.$$

Recall h_L is the mesh size.

[a] Tuo, Wu, and Yu (2014). Surrogate modeling of computer experiments with different mesh densities. *Technometrics*



Determine the fidelity level L

- Let $h_l = h_0 2^{-l}$ where $h_0/2$ is the mesh size of the lowest fidelity simulator $f_1(\mathbf{x})$.

Determine the fidelity level L

- Let $h_l = h_0 2^{-l}$ where $h_0/2$ is the mesh size of the lowest fidelity simulator $f_1(\mathbf{x})$.
- Suppose $|f_\infty(\mathbf{x}) - f_L(\mathbf{x})| = c_1(\mathbf{x}) h_L^\alpha + O(h_L^{\alpha+1})$

Determine the fidelity level L

- Let $h_l = h_0 2^{-l}$ where $h_0/2$ is the mesh size of the lowest fidelity simulator $f_1(\mathbf{x})$.
- Suppose $|f_\infty(\mathbf{x}) - f_L(\mathbf{x})| = c_1(\mathbf{x}) h_L^\alpha + O(h_L^{\alpha+1})$
- One can show that

$$\|f_\infty - f_L\| = \frac{\|f_L - f_{L-1}\|}{2^\alpha - 1},$$

  assuming that the terms of order $h_L^{\alpha+1}$ and higher can be neglected.
- $\|f_L - f_{L-1}\|$ can be approximated by $\|\hat{Z}_L\|$.

Determine the fidelity level L

- Let $h_l = h_0 2^{-l}$ where $h_0/2$ is the mesh size of the lowest fidelity simulator $f_1(\mathbf{x})$.
- Suppose $|f_\infty(\mathbf{x}) - f_L(\mathbf{x})| = c_1(\mathbf{x}) h_L^\alpha + O(h_L^{\alpha+1})$
- One can show that

$$\|f_\infty - f_L\| = \frac{\|f_L - f_{L-1}\|}{2^\alpha - 1},$$

  assuming that the terms of order $h_L^{\alpha+1}$ and higher can be neglected.
- $\|f_L - f_{L-1}\|$ can be approximated by $\|\hat{Z}_L\|$.
- Find L that ensures $\frac{\|\hat{Z}_L\|}{2^\alpha - 1} \leq \epsilon/2$

Determination of α

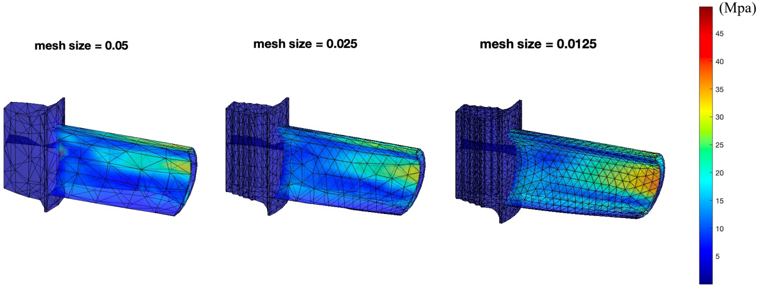
- α can be determined by collected data (can be done only when $L \geq 3$) (details omitted)

$$\hat{\alpha} = \frac{1}{L-2} \sum_{l=3}^{L} \sum_{\mathbf{x} \in X_l} \frac{\log\left(\left|\frac{f_{l-1}(\mathbf{x}) - f_{l-2}(\mathbf{x})}{f_l(\mathbf{x}) - f_{l-1}(\mathbf{x})}\right|\right)}{n_l \log 2}.$$

Questions

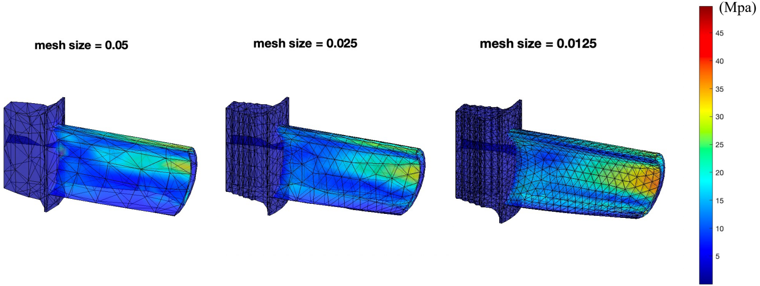
- Q1: How to emulate the exact solution, i.e., $f_\infty(\mathbf{x})$? $\hat{f}_L$
- Q2: Sample size of each level? n_l
- Q3: How many fidelity levels? L
- Q4: Mesh size/density specification? $h_l = h_0 2^{-l}$
- Q5: Is it better than single-fidelity simulation?

Revisit Motivated Example



- **Input**: $\mathbf{x} = (x_1, x_2) = (\text{pressure}, \text{suction})$
- **Output**: $f(\mathbf{x})$: average of thermal stress

Revisit Motivated Example



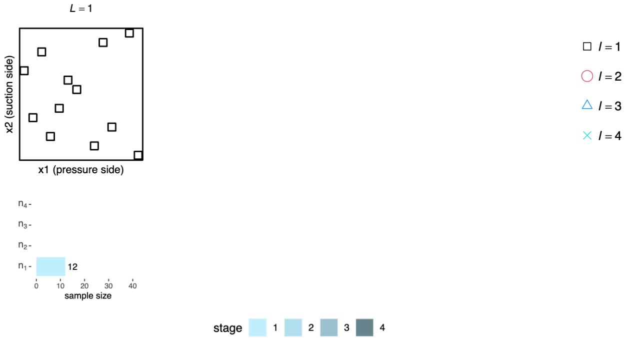
- **Input**: $\mathbf{x} = (x_1, x_2) = (\text{pressure}, \text{suction})$
- **Output**: $f(\mathbf{x})$: average of thermal stress
- **Test data**: Simulations with mesh size $h \approx 0$ at 20 uniform test input locations are conducted to examine the performance

Revisit Motivated Example

- We wish $\|f_\infty - \hat{f}_L\|_{L_2(\Omega)} < \epsilon = 5$

Revisit Motivated Example

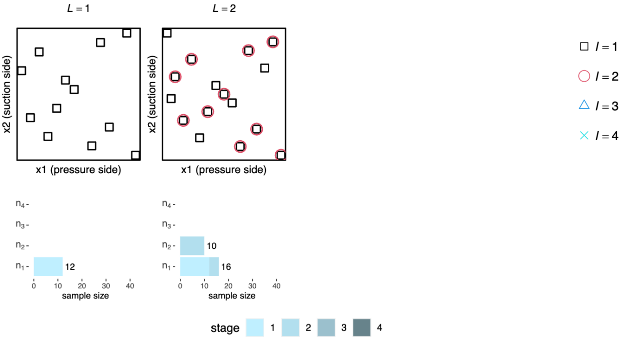
- We wish $\|f_\infty - \hat{f}_L\|_{L_2(\Omega)} < \epsilon = 5$



	$L = 1$
h_L	0.05
C_L (sec.)	0.75
$\|f_L - \hat{f}_L\|_{L_2(\Omega)}$	**2.324**
$\|f_\infty - f_L\|_{L_2(\Omega)}$	NA

Revisit Motivated Example

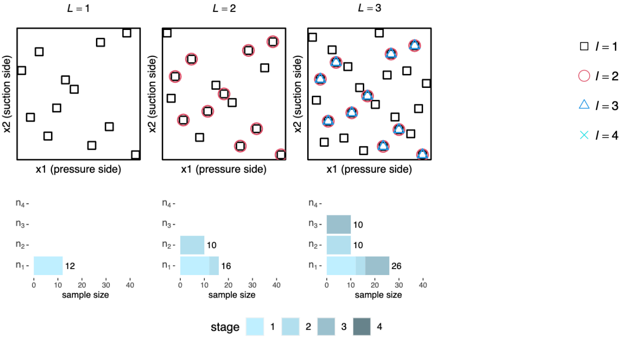
- We wish $\|f_\infty - \hat{f}_L\|_{L_2(\Omega)} < \epsilon = 5$



	$L=1$	$L=2$
h_L	0.05	0.025
C_L (sec.)	0.75	1.07
$\|f_L - \hat{f}_L\|_{L_2(\Omega)}$	**2.324**	**2.408**
$\|f_\infty - f_L\|_{L_2(\Omega)}$	NA	NA

Revisit Motivated Example

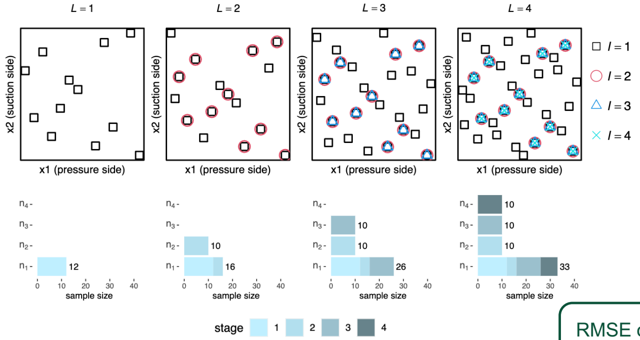
- We wish $\|f_\infty - \hat{f}_L\|_{L_2(\Omega)} < \epsilon = 5$



	$L=1$	$L=2$	$L=3$
h_L	0.05	0.025	0.0125
C_L (sec.)	0.75	1.07	2.13
$\|f_L - \hat{f}_L\|_{L_2(\Omega)}$	**2.324**	**2.408**	**2.481**
$\|f_\infty - f_L\|_{L_2(\Omega)}$	NA	NA	2.969

Revisit Motivated Example

- We wish $\|f_\infty - \hat{f}_L\|_{L_2(\Omega)} < \epsilon = 5$



RMSE of $\hat{f}_4$ $= 1.60$

	$L = 1$	$L = 2$	$L = 3$	$L = 4$	
h_L	0.05	0.025	0.0125	0.00625	
C_L (sec.)	0.75	1.07	2.13	11.51	
$\|f_L - \hat{f}_L\|_{L_2(\Omega)}$	**2.324**	**2.408**	**2.481**	**2.491**	$< \frac{\epsilon}{2}$
$\|f_\infty - f_L\|_{L_2(\Omega)}$	NA	NA	2.969	**0.956**	$< \frac{\epsilon}{2}$

Visualize $\hat{f}_L(\mathbf{x})$

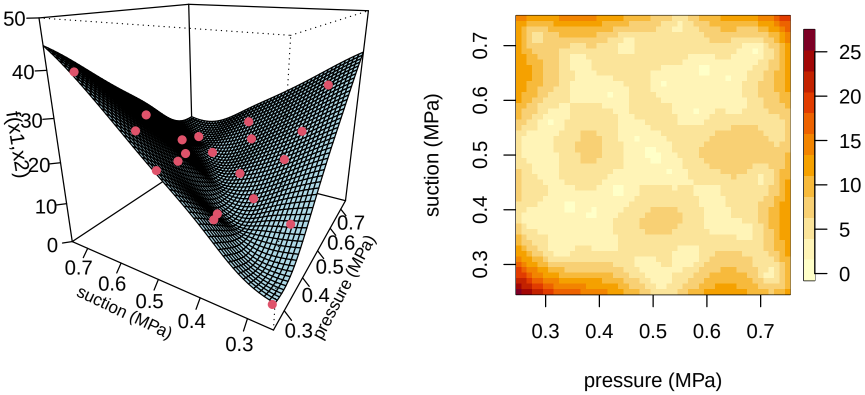


Figure: *(left)* $\hat{f}_4(\mathbf{x})$ *and true test points (red dots); (right) pointwise error bounds*

Cost Complexity Theorem

Theorem

Suppose that

- $\nu := \nu_1 = \cdots = \nu_L$
- $|f_\infty(\mathbf{x}) - f_l(\mathbf{x})| < c_1 2^{-\alpha l}$
- $C_l < c_2 2^{\beta l}$

Cost Complexity Theorem

Theorem

Suppose that

- $\nu := \nu_1 = \cdots = \nu_L$
- $|f_\infty(\mathbf{x}) - f_l(\mathbf{x})| < c_1 2^{-\alpha l}$
- $C_l < c_2 2^{\beta l}$

Under some regularity conditions, it follows that

$$|f_\infty(\mathbf{x}) - \hat{f}_L(\mathbf{x})| < \epsilon,$$

with a **total computational cost** C_{tot} bounded by

$$C_{\text{tot}} \leq \begin{cases} c_3 \epsilon^{-\frac{d}{\nu}}, & \frac{\alpha}{\beta} > \frac{2\nu}{d}, \\ c_3 \epsilon^{-\frac{d}{\nu}} |\log \epsilon|^{1+\frac{d}{\nu}}, & \frac{\alpha}{\beta} = \frac{2\nu}{d}, \\ c_3 \epsilon^{-\frac{d}{\nu} - \frac{2\beta\nu - \alpha d}{2\alpha(\nu+d)}}, & \frac{\alpha}{\beta} < \frac{2\nu}{d}. \end{cases}$$

Complexity of Single-Fidelity Interpolator

Corollary

- Let $\hat{f}_H(x)$ be the RKHS interpolator based on single-fidelity data $(X_H, f_H(X_H))$

Complexity of Single-Fidelity Interpolator

Corollary

- Let $\hat{f}_H(x)$ be the RKHS interpolator based on single-fidelity data $(X_H, f_H(X_H))$
- Suppose that $(\epsilon/2)^{1+\frac{\alpha d}{2\nu\beta}} \leq c_1 h_H^{\alpha} \leq \epsilon/2$, where $c_1 = \sup_{\mathbf{x}\in\Omega} c_1(\mathbf{x})$

Complexity of Single-Fidelity Interpolator

Corollary

- Let $\hat{f}_H(x)$ be the RKHS interpolator based on single-fidelity data $(X_H, f_H(X_H))$
- Suppose that $(\epsilon/2)^{1+\frac{\alpha d}{2\nu\beta}} \leq c_1 h_H^\alpha \leq \epsilon/2$, where $c_1 = \sup_{\mathbf{x}\in\Omega} c_1(\mathbf{x})$

Under some regularity conditions, it follows that

$$|f_\infty(\mathbf{x}) - \hat{f}_H(\mathbf{x})| < \epsilon,$$

with a **total computational cost** C_H bounded by

$$C_H \leq c_h \epsilon^{-\frac{\beta}{\alpha}-\frac{d}{2\nu}}.$$

Single-fidelity vs Multi-fidelity

- When $\frac{\alpha}{\beta} < \frac{2\nu}{d} \Rightarrow$ ML interpolator has a slower cost rate

Single-fidelity vs Multi-fidelity

- When $\frac{\alpha}{\beta} < \frac{2\nu}{\boldsymbol{d}} \Rightarrow$ ML interpolator has a slower cost rate
- When $\frac{\alpha}{\beta} \geq \frac{2\nu}{\boldsymbol{d}} \Rightarrow$ Single-fidelity RKHS interpolator has a slower cost rate

Single-fidelity vs Multi-fidelity

- When $\frac{\alpha}{\beta} < \frac{2\nu}{d} \Rightarrow$ ML interpolator has a slower cost rate
- When $\frac{\alpha}{\beta} \geq \frac{2\nu}{d} \Rightarrow$ Single-fidelity RKHS interpolator has a slower cost rate
- Example 1: β is small and α is large

Single-fidelity vs Multi-fidelity

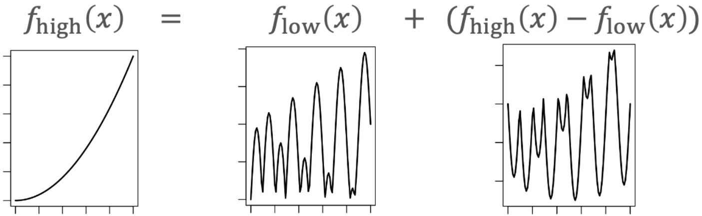
- When $\frac{\alpha}{\beta} < \frac{2\nu}{d} \Rightarrow$ ML interpolator has a slower cost rate
- When $\frac{\alpha}{\beta} \geq \frac{2\nu}{d} \Rightarrow$ Single-fidelity RKHS interpolator has a slower cost rate
- Example 1: β is small and α is large
  - $C_1 = 2.9$ and $C_5 = 3$
  - $|f_\infty(\mathbf{x}) - f_1(\mathbf{x})| = 10$, and $|f_\infty(\mathbf{x}) - f_5(\mathbf{x})| = 0.001$

Single-fidelity vs Multi-fidelity

- When $\frac{\alpha}{\beta} < \frac{2\nu}{d} \Rightarrow$ ML interpolator has a slower cost rate
- When $\frac{\alpha}{\beta} \geq \frac{2\nu}{d} \Rightarrow$ Single-fidelity RKHS interpolator has a slower cost rate
- Example 1: β is small and α is large
  - $C_1 = 2.9$ and $C_5 = 3$
  - $|f_\infty(\mathbf{x}) - f_1(\mathbf{x})| = 10$, and $|f_\infty(\mathbf{x}) - f_5(\mathbf{x})| = 0.001$
- Example 2: ν is very small (i.e., nonsmooth $(f_l - f_{l-1})$)

Single-fidelity vs Multi-fidelity

- When $\frac{\alpha}{\beta} < \frac{2\nu}{d} \Rightarrow$ ML interpolator has a slower cost rate
- When $\frac{\alpha}{\beta} \geq \frac{2\nu}{d} \Rightarrow$ Single-fidelity RKHS interpolator has a slower cost rate
- Example 1: β is small and α is large
  - $C_1 = 2.9$ and $C_5 = 3$
  - $|f_\infty(\mathbf{x}) - f_1(\mathbf{x})| = 10$, and $|f_\infty(\mathbf{x}) - f_5(\mathbf{x})| = 0.001$
- Example 2: ν is very small (i.e., nonsmooth $(f_l - f_{l-1})$)



Questions

- Q1: How to emulate the exact solution, i.e., $f_\infty(\mathbf{x})$? $\hat{f}_L$
- Q2: Sample size of each level? n_l
- Q3: How many fidelity levels? L
- Q4: Mesh size/density specification? $h_l = h_0 2^{-l}$
- Q5: Is it better than single-fidelity simulation? In some cases, yes, but not always

Recall Our Model

- **ML Interpolator**:

$$Z_l(\mathbf{x}) = f_l(\mathbf{x}) - f_{l-1}(\mathbf{x}), \quad l = 2, \ldots, L.$$

- How about **Auto-Regressive (AR) model**[5]:

$$f_l(\mathbf{x}) = \rho_{l-1} f_{l-1}(\mathbf{x}) + Z_{l-1}(\mathbf{x}), \quad l = 2, \ldots, L$$

where both $f_{l-1}(\mathbf{x})$ and $Z_{l-1}(\mathbf{x})$ follow Gaussian Process (GP) priors.

[5] Kennedy and O'Hagan (2000). Predicting the output from a complex computer code when fast approximations are available. *Biometrika*

Recall Our Model

- **ML Interpolator**:

$$Z_l(\mathbf{x}) = f_l(\mathbf{x}) - f_{l-1}(\mathbf{x}), \quad l = 2, \ldots, L.$$

- How about **Auto-Regressive (AR) model**[5]:

$$f_l(\mathbf{x}) = \rho_{l-1} f_{l-1}(\mathbf{x}) + Z_{l-1}(\mathbf{x}), \quad l = 2, \ldots, L$$

  where both $f_{l-1}(\mathbf{x})$ and $Z_{l-1}(\mathbf{x})$ follow Gaussian Process (GP) priors.

- Both rely on an *additive* (or *linear*) structure.

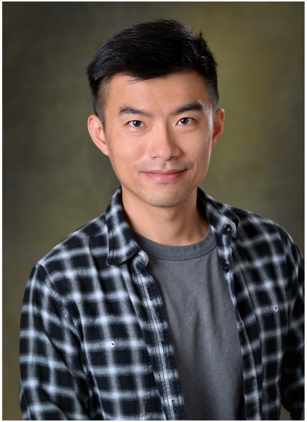
[5] Kennedy and O'Hagan (2000). Predicting the output from a complex computer code when fast approximations are available. *Biometrika*

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

Active learning for a recursive non-additive emulator for multi-fidelity computer experiments, *Technometrics*, to appear.

MICHIGAN STATE UNIVERSITY

Junoh Heo



Chih-Li Sung

Home ▶ All Journals ▶ Technometrics

Technometrics

Enter keywords, authors, DOI, etc

Submit an article | About this journal | Browse all articles & issues | Alerts & RSS feed

Explore articles

Latest | Open access | **Most read** | Most cited | Trending

Browse the most read articles published in the last year

Discrepancy Measures for Global Sensitivity Analysis ›
Arnald Puy et al.
Article | Published online: 13 Feb 2024
1362 Views

Active Learning for a Recursive Non-Additive Emulator for Multi-Fidelity Computer Experiments ›
Junoh Heo et al.
Article | Published online: 17 Sep 2024
1326 Views

Constrained Bayesian Optimization with Lower Confidence Bound ›
Neelesh S. Upadhye et al.
Article | Published online: 7 May 2024
1186 Views

Data Analytics for Process Engineers; Prediction, Control, and Optimization ›
Eduwin Saputra et al.
Book Review | Published online: 1 May 2024
1083 Views

Mesh-Clustered Gaussian Process Emulator for Partial Differential Equation Boundary Value Problems ›
Chih-Li Sung et al.
Article | Published online: 28 Mar 2024
962 Views

Existing Methods

- Q: Would it always follow an additive structure?

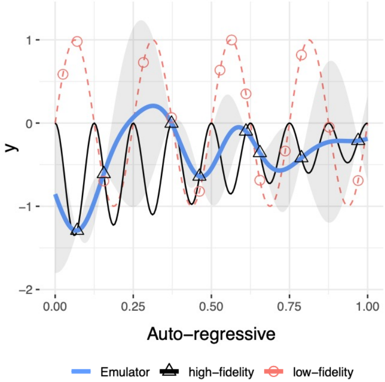


Figure: *A synthetic example*[6]*, where* $n_1 = 13$, $n_2 = 8$, $f_1(x) = \sin(8\pi x)$*, and* $f_2(x) = (x - \sqrt{2})f_1^2(x)$.

[6] Perdikari et al. (2017)

RNA Emulator

- We propose a Recursive Non-Additive emulator (RNA emulator) to overcome this limitation in a recursive fashion:

$$f_1(\mathbf{x}) = W_1(\mathbf{x}),$$

$$f_l(\mathbf{x}) = W_l(\mathbf{x}, f_{l-1}(\mathbf{x})), \quad l = 2, \cdots, L,$$

- The auto-regressive model ($f_l(\mathbf{x}) = \rho_{l-1} f_{l-1}(\mathbf{x}) + Z_l(\mathbf{x})$) becomes a special case!

RNA Emulator

- We propose a Recursive Non-Additive emulator (RNA emulator) to overcome this limitation in a recursive fashion:

$$f_1(\mathbf{x}) = W_1(\mathbf{x}),$$

$$f_l(\mathbf{x}) = W_l(\mathbf{x}, f_{l-1}(\mathbf{x})), \quad l = 2, \cdots, L,$$

- The auto-regressive model ($f_l(\mathbf{x}) = \rho_{l-1} f_{l-1}(\mathbf{x}) + Z_l(\mathbf{x})$) becomes a special case!
- Model the relationship $\{W_l\}_{l=1}^{L}$ using independent GP priors

RNA Emulator

RNA Emulator

$$W_1(\mathbf{x}) \sim \mathcal{GP}(\alpha_1, \tau_1^2 \Phi_1(\mathbf{x}, \mathbf{x}')),$$

$$W_l(\mathbf{z}) \sim \mathcal{GP}(\alpha_l, \tau_l^2 K_l(\mathbf{z}, \mathbf{z}')), \quad l = 2, \cdots, L,$$

where $\mathbf{z} = (\mathbf{x}, y)$, and $\Phi_1(\mathbf{z}, \mathbf{z}')$ and $K_l(\mathbf{z}, \mathbf{z}')$ are a positive definite kernel.

RNA Emulator

RNA Emulator

$$W_1(\mathbf{x}) \sim \mathcal{GP}(\alpha_1, \tau_1^2 \Phi_1(\mathbf{x}, \mathbf{x}')),$$

$$W_l(\mathbf{z}) \sim \mathcal{GP}(\alpha_l, \tau_l^2 K_l(\mathbf{z}, \mathbf{z}')), \quad l = 2, \cdots, L,$$

where $\mathbf{z} = (\mathbf{x}, y)$, and $\Phi_1(\mathbf{z}, \mathbf{z}')$ and $K_l(\mathbf{z}, \mathbf{z}')$ are a positive definite kernel.

- e.g., squared exponential kernel:

$$\Phi_l(\mathbf{x}, \mathbf{x}') = \prod_{j=1}^{d} \exp\left(-\frac{(x_j - x_j')^2}{\theta_{lj}}\right)$$

$$K_l(\mathbf{z}, \mathbf{z}') = \exp\left(-\frac{(y - y')^2}{\theta_{ly}}\right) \prod_{j=1}^{d} \exp\left(-\frac{(x_j - x_j')^2}{\theta_{lj}}\right)$$

Closed Form Expression of RNA Emulator

Proposition 1: The closed-form expressions

- Under the **squared exponential kernel**, the posterior mean and variance can be obtained as follows:

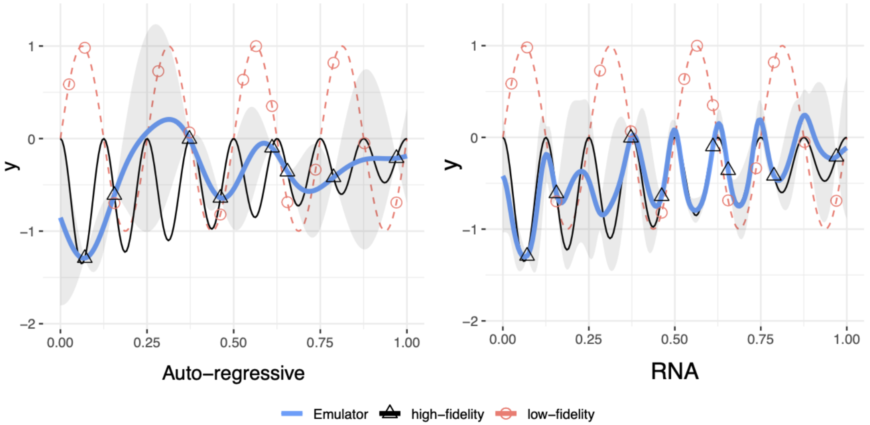
$$\mu_l^*(\mathbf{x}) := \mathbb{E}[f_l(\mathbf{x})|\mathbf{y}_1,\ldots,\mathbf{y}_l]$$

$$= \alpha_l + \sum_{i=1}^{n_l} r_i \prod_{j=1}^{d} \exp\left(-\frac{(x_j - x_{ij}^{[l]})^2}{\theta_{lj}}\right) \frac{1}{\sqrt{1+2\frac{\sigma_{l-1}^{*2}(\mathbf{x})}{\theta_{ly}}}} \exp\left(-\frac{(y_i^{[l-1]} - \mu_{l-1}^*(\mathbf{x}))^2}{\theta_{ly} + 2\sigma_{l-1}^{*2}(\mathbf{x})}\right),$$

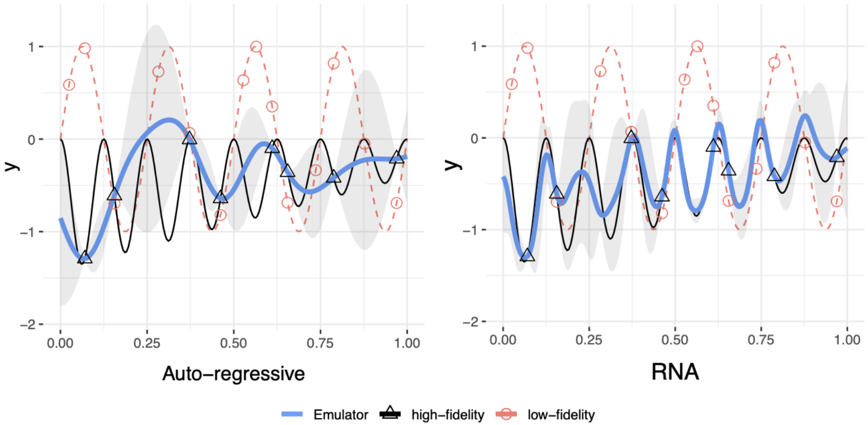
$$\sigma_l^{*2}(\mathbf{x}) := \mathbb{V}[f_l(x)|\mathbf{y}_1,\ldots,\mathbf{y}_l] = \tau_l^2 - (\mu_l^*(\mathbf{x}) - \alpha_l)^2 +$$

$$\left(\sum_{i,k=1}^{n_l} \zeta_{ik}\left(r_i r_k - \tau_l^2(\mathbf{K}_l^{-1})_{ik}\right) \prod_{j=1}^{d} \exp\left(-\frac{(x_j - x_{ij}^{[l]})^2 + (x_j - x_{kj}^{[l]})^2}{\theta_{lj}}\right)\right).$$

RNA Emulator



After emulating...



However, the emulator still holds the uncertainty in some regions!

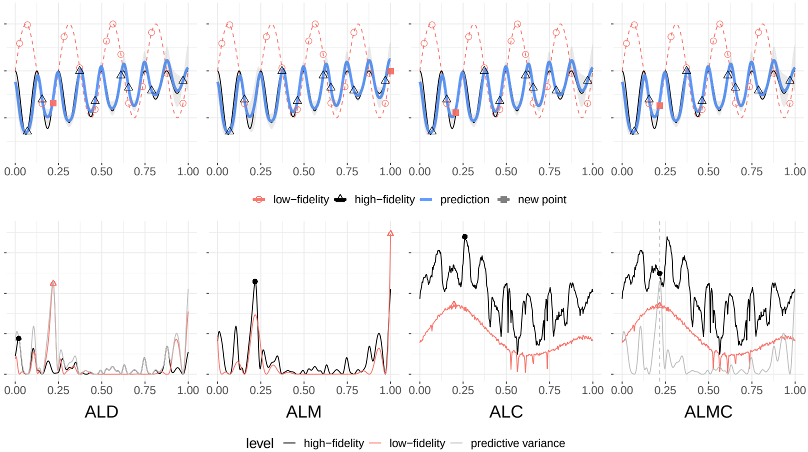
Active Learning for RNA emulator

- In multi-fidelity simulation, active learning requires
  - identifying optimal input locations,
  - identifying fidelity levels,
  - accounting for the respective simulation costs simultaneously.

Active Learning for RNA emulator

- In multi-fidelity simulation, active learning requires
  - identifying optimal input locations,
  - identifying fidelity levels,
  - accounting for the respective simulation costs simultaneously.
- Four active learning strategies for RNA emulator will be introduced: ALD, ALM, ALC, and ALMC.

Active Learning for RNA Emulator



Revisit Motivated Example (Blade Simulation)

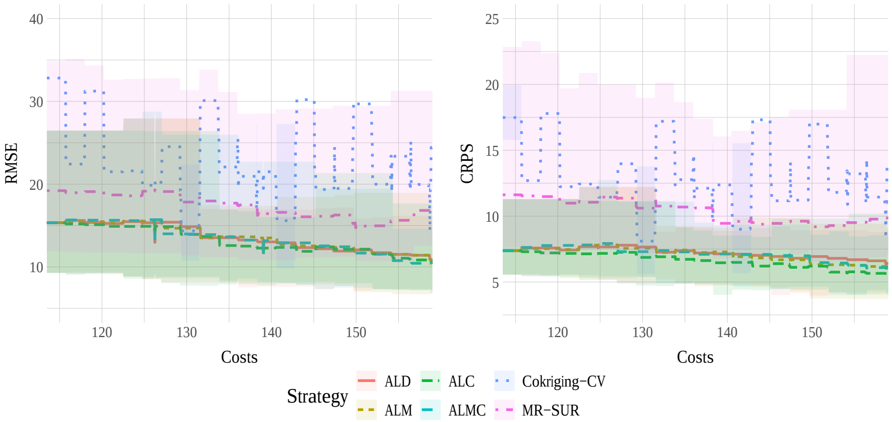


Figure: *RMSE and CRPS for the motivated example with respect to the cost.*

Conclusion

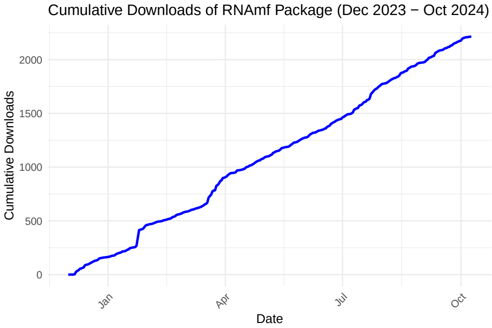
- **Stacking Designs**:
  - Emulates $f_\infty(\mathbf{x})$ with theoretical guarantees.
  - Answers key questions, such as optimal sample size and the number of fidelity levels.
  - Provides insights into the comparison between single-fidelity and multi-fidelity approaches.

Conclusion

- **Stacking Designs**:
  - Emulates $f_\infty(\mathbf{x})$ with theoretical guarantees.
  - Answers key questions, such as optimal sample size and the number of fidelity levels.
  - Provides insights into the comparison between single-fidelity and multi-fidelity approaches.
- **Active Learning for RNA Emulator**:
  - A more flexible model for emulating $f_L(\mathbf{x})$.
  - Flexibility comes without additional computational cost due to closed-form posterior mean and variance expressions.
  - Four active learning strategies are introduced to select fidelity level and sample location, enhancing emulation accuracy.

Open-Source Contributions

- R package RNAmf (over 2,200 downloads) is available.



- Reproducibility code for both papers is available on GitHub.

Acknowledgement

- **National Science Foundation**
  - NSF DMS 2338018: 2024-2029 (PI, $423,591)
    CAREER: Single-Fidelity vs. Multi-Fidelity Computer Experiments: Unveiling the Effectiveness of Multi-Fidelity Emulation
  - NSF DMS 2113407: 2021-2024 (PI, $142,009)
    Collaborative Research: Efficient Bayesian Global Optimization with Applications to Deep Learning and Computer Experiments

Thank You!

MICHIGAN STATE
UNIVERSITY