Active Learning for Finite Element Simulations with Adaptive Non-Stationary Kernel Function

Romain Boutelet and Chih-Li Sung

Department of Statistics and Probability
Michigan State University

ASA/IMS Spring Research Conference 2025

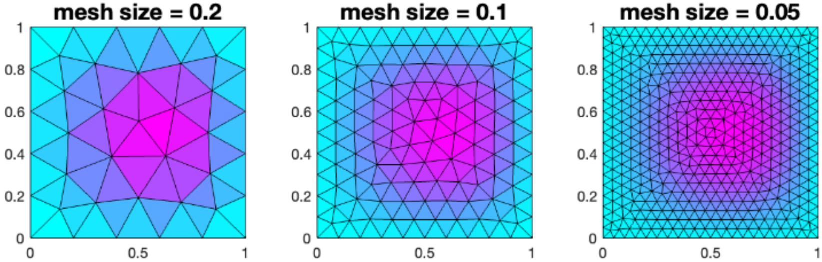
Table of Contents

Motivation

- Computer experiments are widely used to solve **complex** mathematical models.
- They can however be prohibitively **expensive** to explore a full parameter space.
- Multi-fidelity experiments leverage the correlation between fidelity levels (ie. mesh sizes) to obtain a **surrogate model** on the whole parameter space.

Motivating example: Finite Element Methods

There is a trade-off between **accuracy** and **computational cost**.



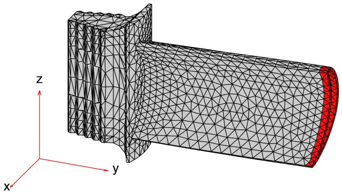
cheap — cost → — expensive

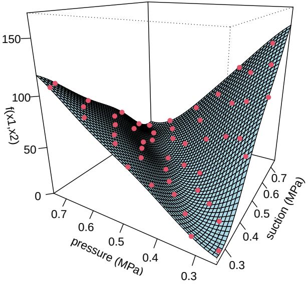
low accuracy — ← accuracy — high accuracy

The figure is from Sung et al. (2022)

Setting of Multi-Fidelity Simulations

- Two types of inputs:
  - **Input parameter**, which can be control variables, environmental variables, or calibration variables.
  - **Mesh size**, which determine the accuracy of the numerical computations.
- The **output** must be a scalar.





Multi-Fidelity Emulators

Objective:

Use information from different **mesh sizes** to make accurate predictions of the **true solution** for any **input parameter**.

Existing methods:

- Auto-regressive model (O'Hara and Kennedy, 2000)

  $f_l(\mathbf{x}) = \rho_{l-1} f_{l-1}(\mathbf{x}) + Z_l(\mathbf{x})$ $\rightarrow$ finite number of fidelity levels

- Non-stationnary model (Tuo et al., 2014)

  Idea: integrating the **infinite** potential fidelity levels into an additional dimension of a single GP.

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

Non-Stationary Model

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

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

where $\varphi(\mathbf{x}) := y(\mathbf{x}, 0)$ and $\delta(\mathbf{x}, t)$ are represented as two mutually independent GPs.

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

More in detail

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

$$\mathbb{E}[\varphi(\mathbf{x})] = f_1^T(\mathbf{x})\beta_1, \quad \mathbb{E}[\delta(\mathbf{x}, t)] = f_2^T(\mathbf{x}, t)\beta_2$$

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

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

More in detail

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

$$\mathbb{E}[\varphi(\mathbf{x})] = f_1^T(\mathbf{x})\beta_1, \quad \mathbb{E}[\delta(\mathbf{x}, t)] = f_2^T(\mathbf{x}, t)\beta_2$$

- 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

  - The **true solution** has a stationary covariance function
  $K_1(\mathbf{x}, \mathbf{x}') = \sigma_1^2 \prod_{i=1}^d e^{-\phi_1^2 (x_i - x_i')^2}$
  - The **error function** has a **non-stationary** covariance function
  $K_2(\mathbf{x}, \mathbf{x}', t, t') = \sigma_2^2 K_H(t, t') \prod_{i=1}^d e^{-\phi_2^2 (x_i - x_i')^2}$

Extending the Non-Stationary model

FBM kernel

We introduce a new covariance function on the mesh size t, adapted from the **Fractional Brownian Motion** (FBM).

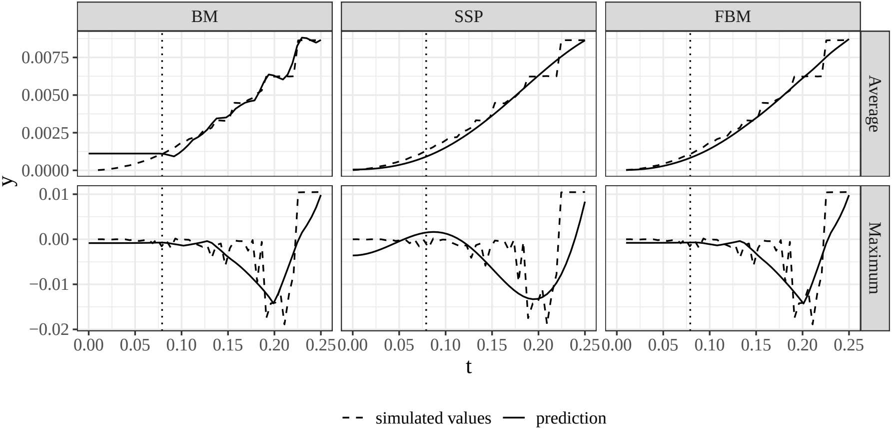
$$F_H(t,t') = \left\{ \frac{1}{2}(t^{2H} + (t')^{2H} + |t-t'|^{2H}) \right\}^{\frac{l}{2H}}, \quad 0 < H < 1$$

<u>**Remark:**</u> Tuo, Wu and Yu (2014) originally proposed two covariance functions:

- The Brownian Motion, $BM(t,t') = \min(t,t')^l$.
- The Scaled Stationary Process, $SSP(t,t') = \left(\sqrt{t,t'}e^{-\phi(t-t')^2}\right)^l$

Illustration of the Covariance Function

The **FBM** kernel provides more flexibility to accommodate different **correlation** levels in between mesh sizes.



- **Active Learning (AL) designs**
  - optimize the choice of the next design point by a given **criterion**,
  - utilize all the available information (model fit, uncertainty,...).

Active Learning and Multi Fidelity Emulators

- **Active Learning (AL) designs**
  - optimize the choice of the next design point by a given **criterion**,
  - utilize all the available information (model fit, uncertainty,...).
- **In the context of our model**
  - The **criterion** must aim at improving the prediction of the true solution.
  - We must take the **computational cost** into account for multi-fidelity simulations.
  - The method will (*hopefully*) show different behaviors depending on the model parameters.

Integrated Mean Square Prediction Error (IMSPE)

- We use the **Integrated Mean Square Prediction Error** (IMSPE) as the foundation of our active learning criterion.
- The IMSPE from the n initial design elements, I_n, can be written as

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

- Here, $\hat{\sigma}_n^2(\mathbf{x}, 0)$ is the **predictive variance** for the exact solution.

Calculating the IMSPE sequentially

- The active learning objective is to find the best new design location $(\mathbf{x}_{n+1}, t_{n+1})$ by minimizing $I_{n+1}(\mathbf{x}, t)$ on the whole space $\mathcal{X} \times \mathcal{T}$.

Theorem: IMSPE reduction

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

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

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 $R_{n+1}(\tilde{\mathbf{x}}, \tilde{t})$ and the cost $C(\tilde{t})$:

$$(\mathbf{x}_{n+1}, t_{n+1}) = \underset{(\mathbf{x},t)\in\mathcal{D}\times\mathcal{T}}{\operatorname{argmax}} \frac{R_{n+1}(\mathbf{x}, t)}{C(t)}$$

- The optimization is performed on the whole space $\mathcal{D} \times \mathcal{T}$ jointly, taking advantage of the whole range of mesh sizes accessible.

Demonstration

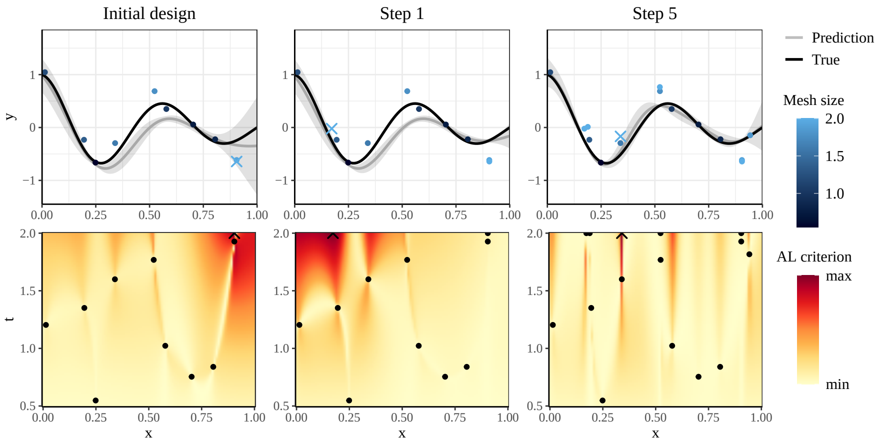


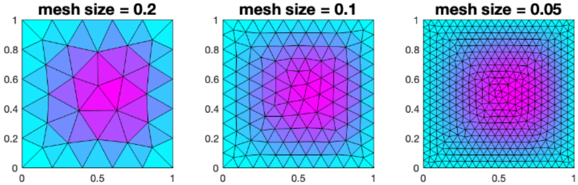
Figure: Prediction of our model against the exact solution (top), and the active learning criterion surface (bottom). The points represent the current design locations (•), and the best next design location according to the criterion ($\times$).

Elliptic PDE

Poisson's equation on a square membrane

$$\Delta u = (x^2 + 2\pi^2)e^{xz_1}\sin(\pi z_1)\sin(\pi z_2) + 2x\pi e^{xz_1}\cos(\pi z_1)\sin(\pi z_2)$$

$$u = 0 \text{ on } \partial\mathcal{D}$$

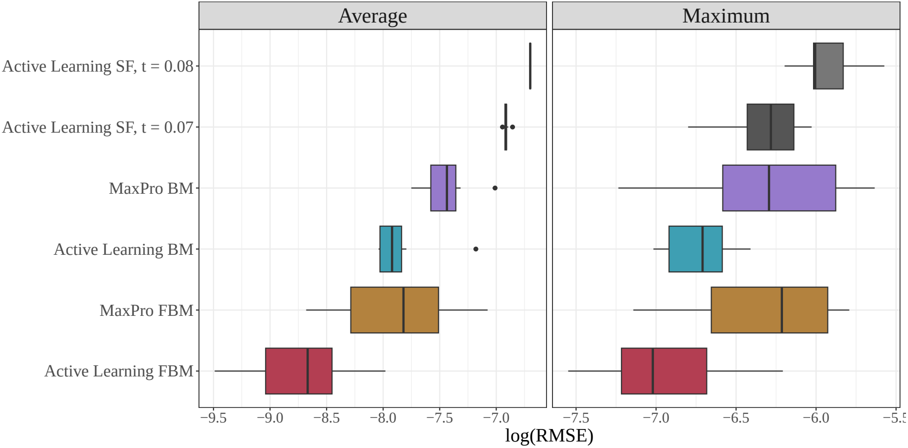


We are interested in two different response of interest:

- the **average** over $\mathcal{D}$, which has a closed form expression.
- the **maximum** over $\mathcal{D}$, which can be accurately approximated.

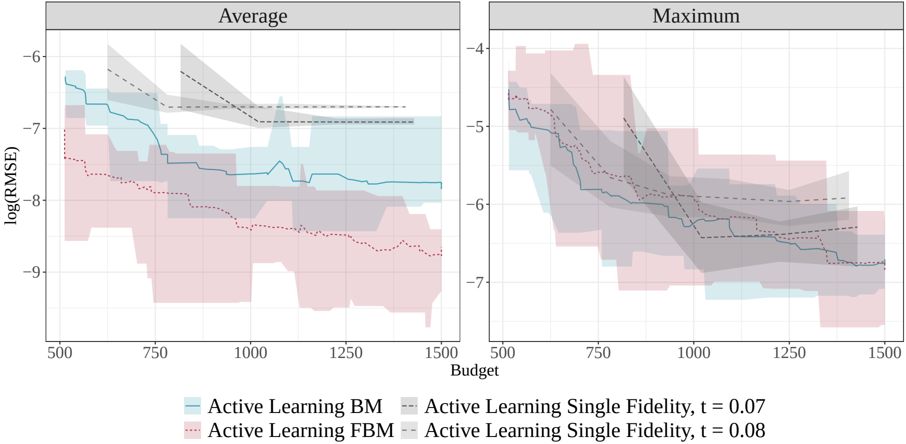
Results

We compare the predictive performance of our model using different kernels (**FBM** vs **BM**) and designs (**Active Learning** vs **MaxPro**). We also compare with methods using only a single fidelity level (**SF**).



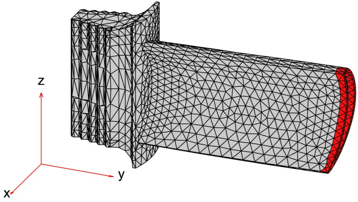
Continued: Results

We now compare the different active learning methods according to their predictive performance according to the **budget** used.

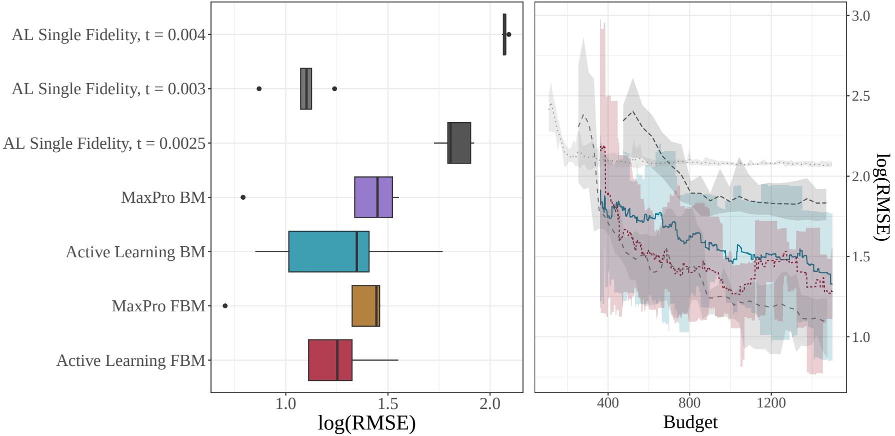


Analysis of a jet engine turbine blade

- We study a static structural problem on a jet engine turbine blade, which we solve using FEM toolbox on `Matlab`.
- We are interested in the **maximum displacement** of the blade tip (in red) in the y-direction.
- There is no true value for evaluating predictive performance, so we compute a set of testing points with **very fine mesh size**.



Results



AL Single Fidelity, t = 0.004
AL Single Fidelity, t = 0.003
AL Single Fidelity, t = 0.0025
MaxPro BM
Active Learning BM
MaxPro FBM
Active Learning FBM
1.0
1.5
2.0
log(RMSE)
400
800
1200
Budget
3.0
2.5
2.0
1.5
1.0
log(RMSE)

Conclusion

What we did:

- We introduced a **novel** and **adaptive** covariance function for the non-stationary model.
- We proposed an **active learning** strategy adapted to the multi-fidelity setting.
- Numerical studies show the **effectiveness** of our method compared other methods.

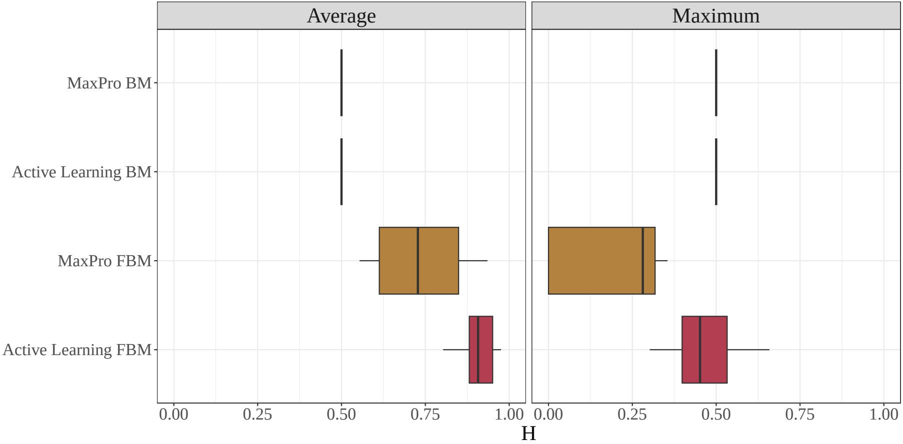
What we need to explore:

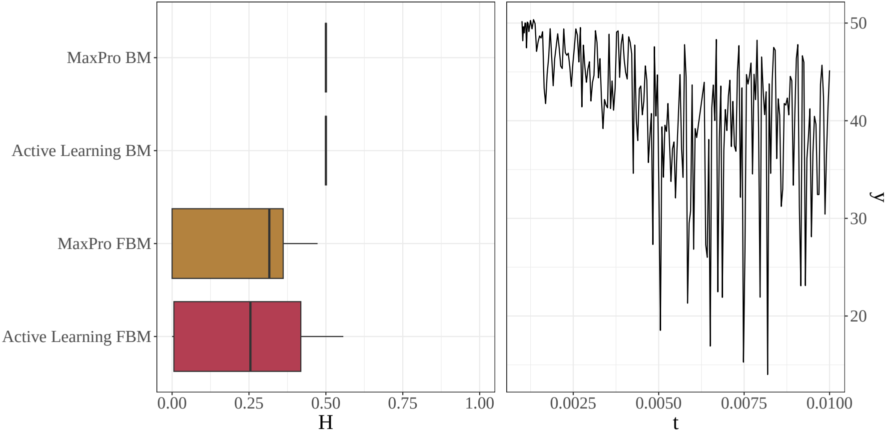
- Developing an **initial design** that gives robust parameter estimation and final prediction.
- Extend the model to a **high-dimensional** output.

Thank You!

Romain Boutelet, Chih-Li Sung (2025). *Active Learning for Finite Element Simulations with Adaptive Non-Stationary Kernel Function.* arXiv

Questions?


Average
Maximum
MaxPro BM
Active Learning BM
MaxPro FBM
Active Learning FBM
0.00
0.25
0.50
0.75
1.00
H


MaxPro BM
Active Learning BM
MaxPro FBM
Active Learning FBM
0.00
0.25
0.50
0.75
1.00
H
50
40
30
20
y
0.0025
0.0050
0.0075
0.0100
t