SAND2023-06739C

Multi-fidelity Surrogate Modeling: A tutorial

J.D. Jakeman, Gianluca Geraci

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525.

Sandia National Laboratories | U.S. Department of Energy | Office of Science

Advanced Scientific Computing Research

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

COLLABORATORS

Left to Right

John Jakeman, Michael Eldred, Gianluca Geraci (SNL)



Tom Smith, Tom Seidl (SNL)
Alex Gorodetsky (University of Michigan)

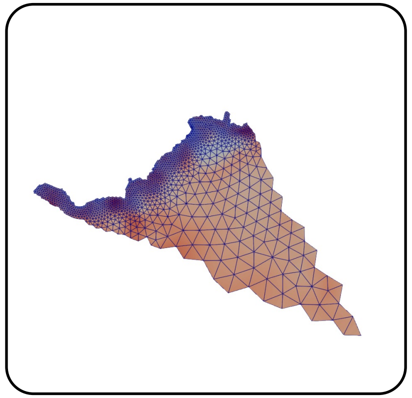


UNCERTAINTY QUANTIFICATION

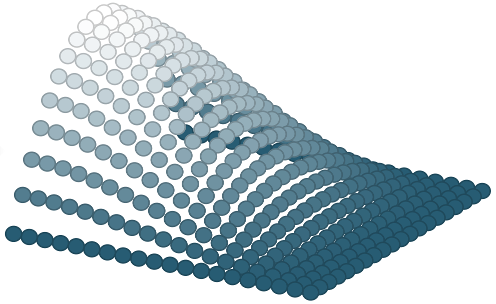
All models are approximations of reality. Sources of uncertainty must be identified and their effect on predictions quantified

Variables
Friction
Surface mass balance
…

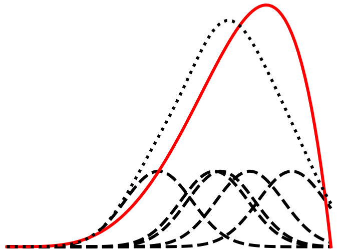
$z = [z_1, \ldots, z_D]$



$u_\alpha(z)$



$f_\alpha(u_\alpha(z))$



$\pi(f_\alpha)$

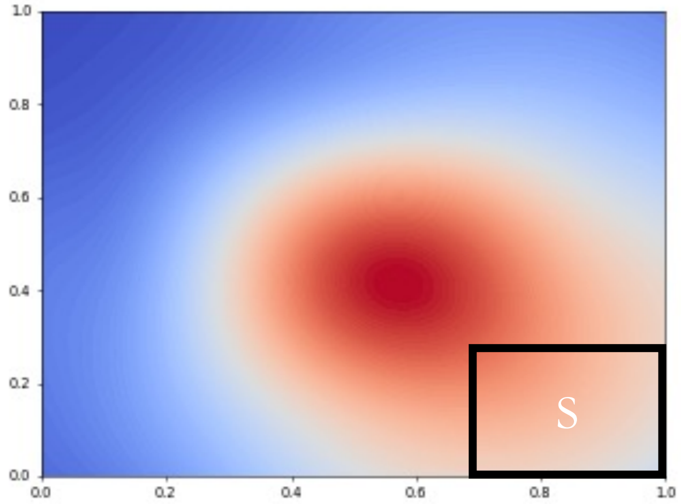
Parameterize Uncertainty

Transient Advection diffusion

$$\frac{\partial u}{\partial t}(x,t,z) = \nabla\cdot[k(x,z)\nabla u(x,t,z)] - \nabla\cdot(vu(x,t,z)) + g(x,t)(x,t,z) \in D\times[0,1]\times\Gamma$$

$$\mathcal{B}(x,t,z) = 0(x,t,z) \in \partial D\times[0,1]\times\Gamma$$

$$u(x,t,z) = u_0(x,z)(x,t,z) \in D\times\{t=0\}\times\Gamma$$

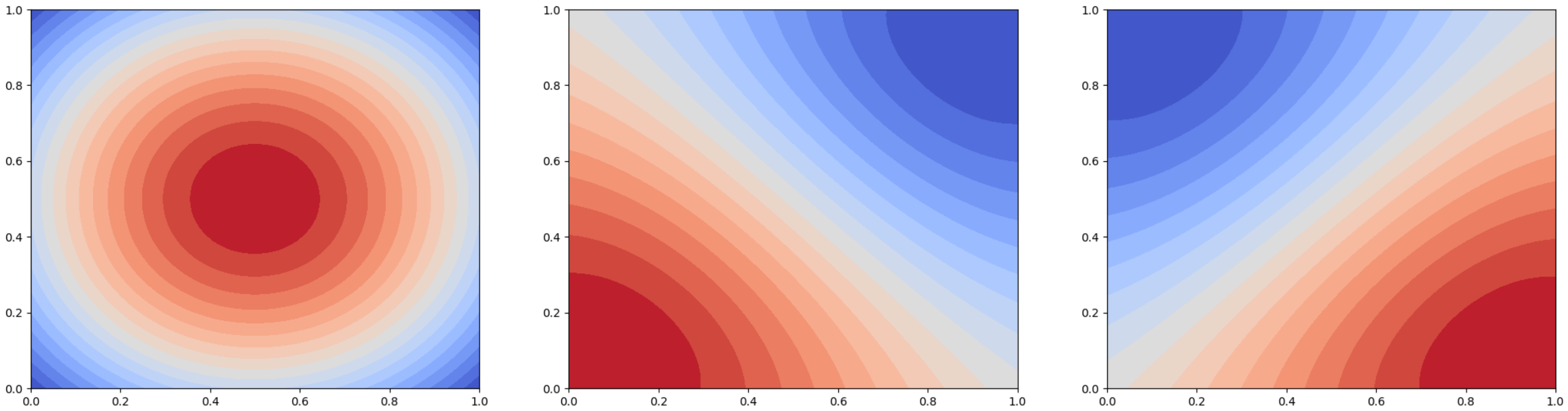


Accuracy of QoI $f(z)$ depends on numerical discretization

$$f(z) = \int_S u(x,T,z)dx$$

Often diffusion field is unknown so parameterize with a Karhunen Loeve expansion (realizations shown below)

$$k(x,z) = \exp\left(\sum_{d=1}^{D}\sqrt{\lambda_d}\psi_d(x)z_d\right)$$

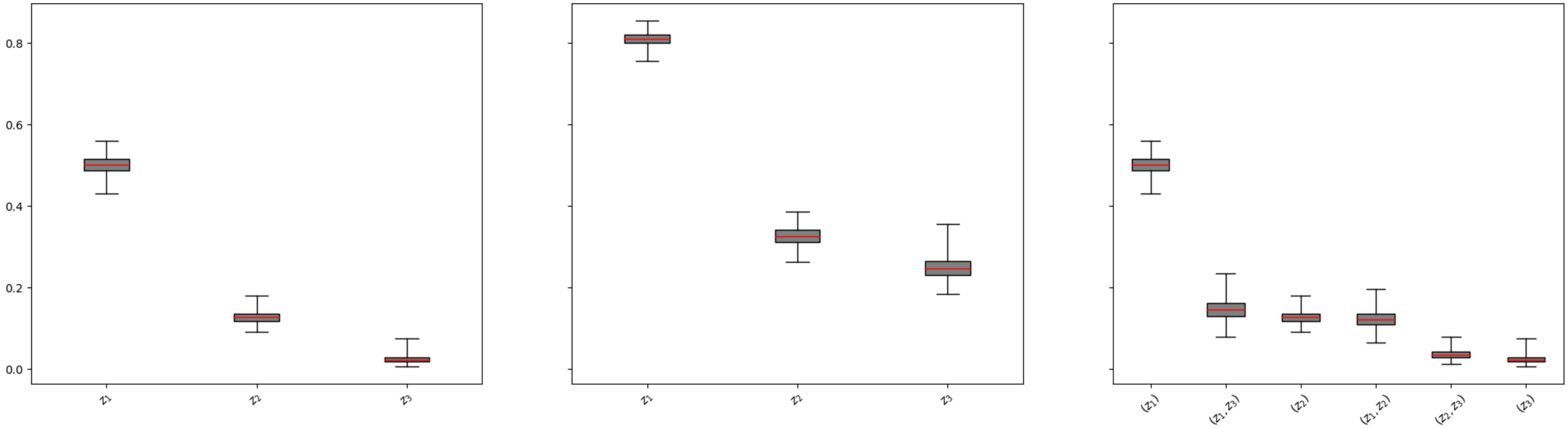


Sensitivity Analysis

Sensitivity Analysis quantifies the impact of variable subsets on predictions

$$f(z) = \hat{f}_0 + \sum_{i=1}^{d} \hat{f}_i(z_i) + \sum_{i,j=1}^{d} \hat{f}_{i,j}(z_i, z_j) + \cdots + \hat{f}_{1,\ldots,d}(z_1, \ldots, z_d)$$

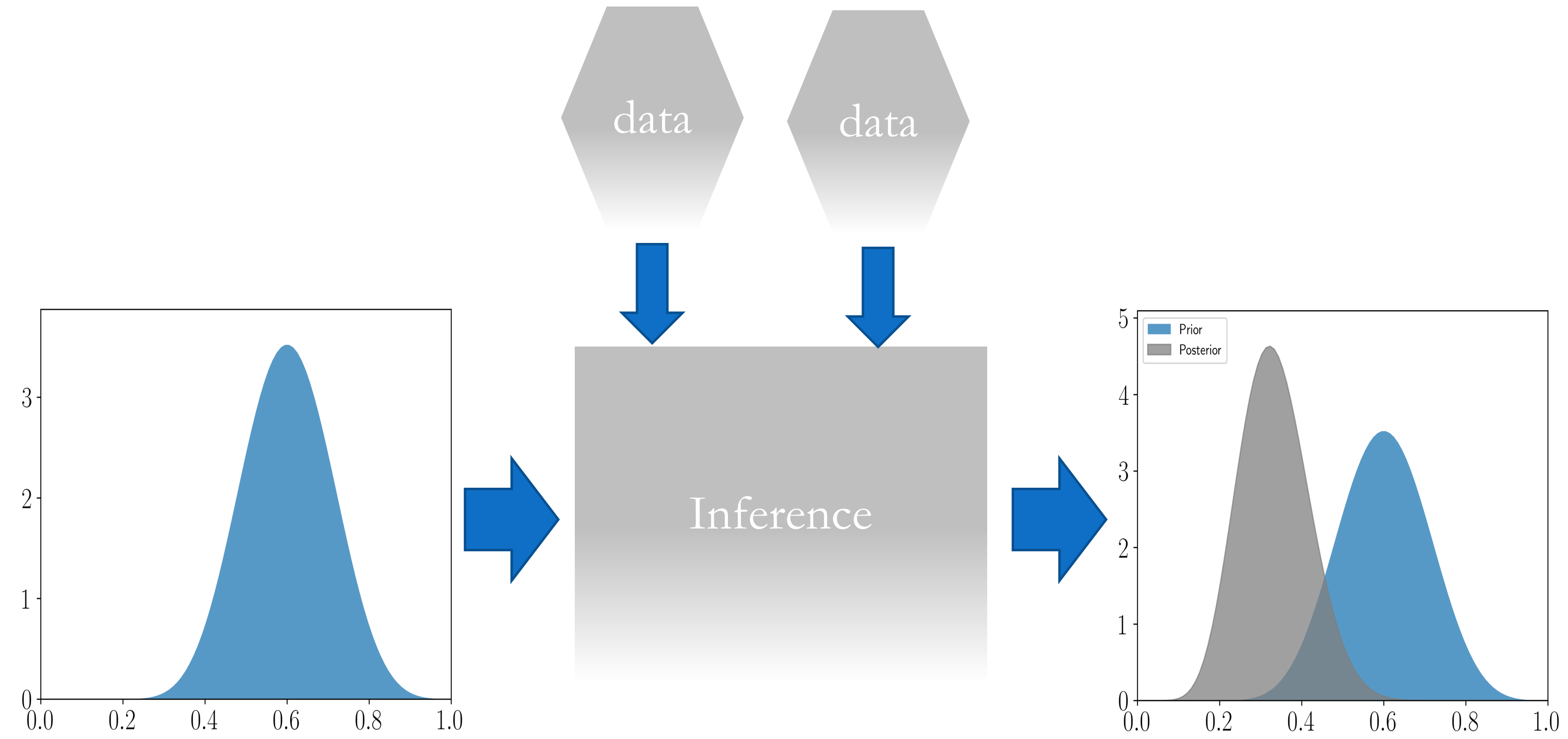
$$S_i = \frac{\mathbb{V}\left[\hat{f}_{\boldsymbol{e}_i}\right]}{\mathbb{V}[f]}, \qquad S_i^T = \frac{\sum_{\boldsymbol{u} \in \mathcal{J}} \mathbb{V}\left[\hat{f}_{\boldsymbol{u}}\right]}{\mathbb{V}[f]}$$



Bayesian inference (calibration)

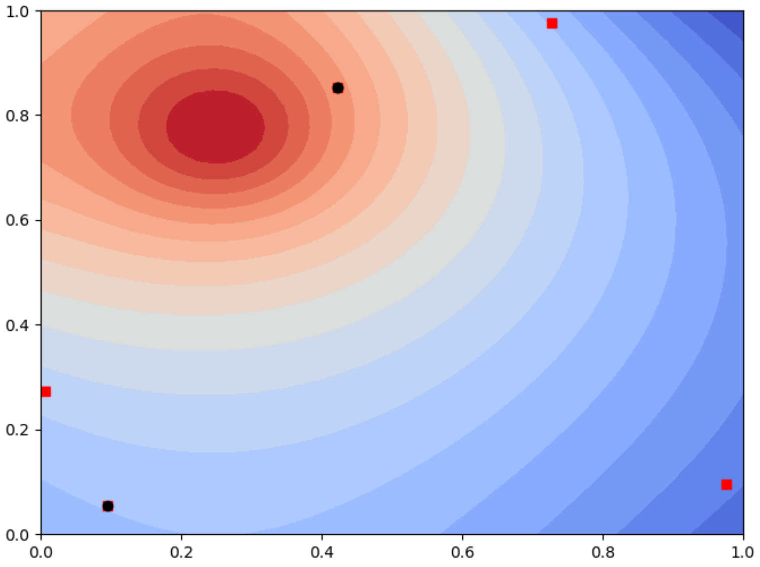
Bayesian inference uses data $\boldsymbol{d}$ to update estimates of uncertainty

$$\pi(z \mid d) = \frac{\pi(d \mid z)\,\pi(z)}{\int_{\mathbb{R}^d} \pi(d \mid z)\,\pi(z)\,dz}$$



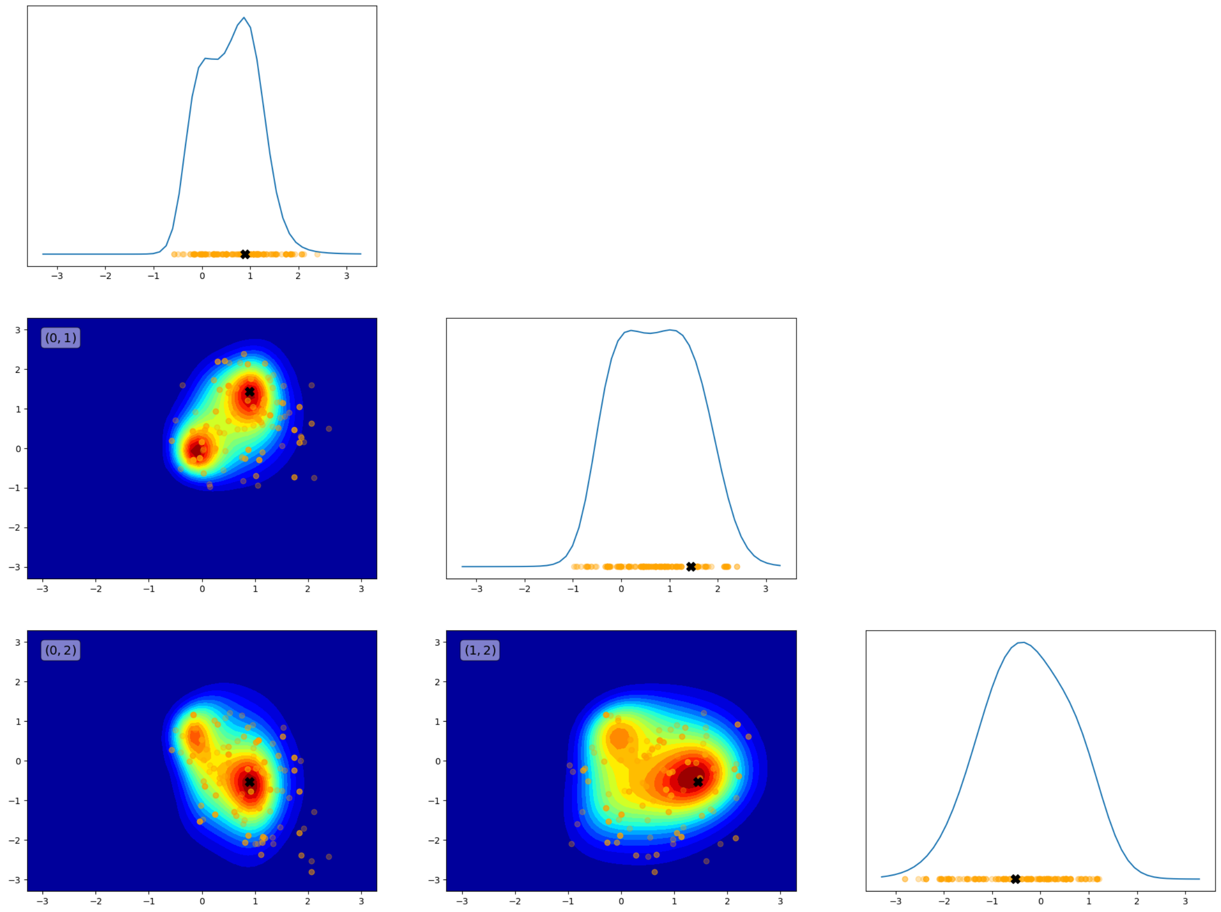
BAYESIAN INFERENCE (CALIBRATION)

We can collect sparse measurements of concentration



Experimental design can be used to choose data that will most inform uncertainty

And update prior distribution on coefficients

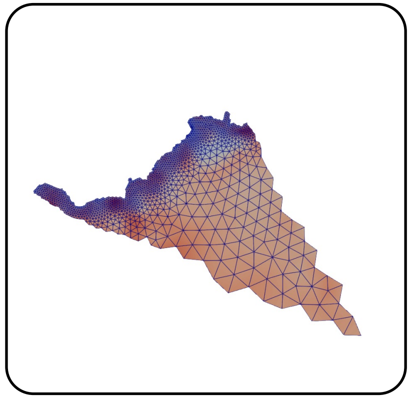


Forward propagation

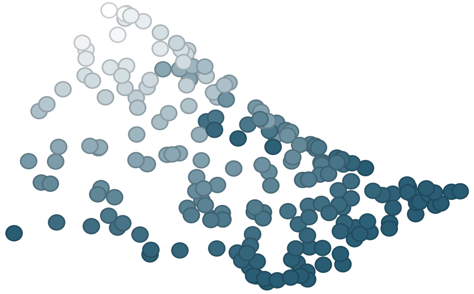
Forward propagation of uncertainty computes measures of prediction uncertainty from a set of model evaluations

Variables
Friction
Surface mass balance
…

$z = [z_1, \ldots, z_D]$



$u_\alpha(z)$



$f_\alpha(u_\alpha(z))$

$$\mathbb{E}[f_\alpha(z)] + \mathbb{V}[f_\alpha(z)]^{1/2}$$

$$\mathbb{P}[f_\alpha(z) \geq \delta]$$

When a model is computational expensive only a limited number of evaluations may be available.

UNCERTAINTIES EFFECT DESIGNS

Deterministic design

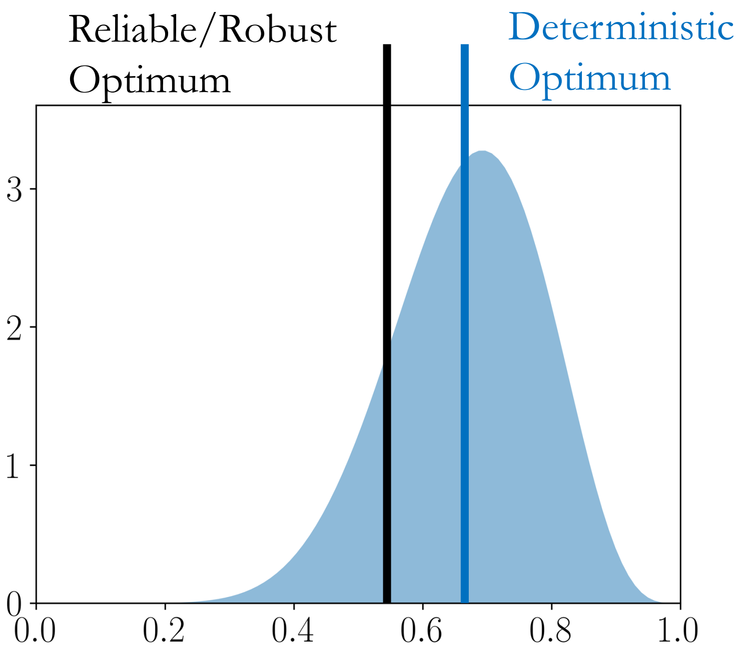
Compute design at nominal values $z^\star$

$$\min_{\xi\in\Xi} \mathcal{L}\big(u(z^\star,\xi)\big)$$
$$s.t.\ \ \mathcal{C}\big(u(z^\star,\xi)\big) \leq 0$$

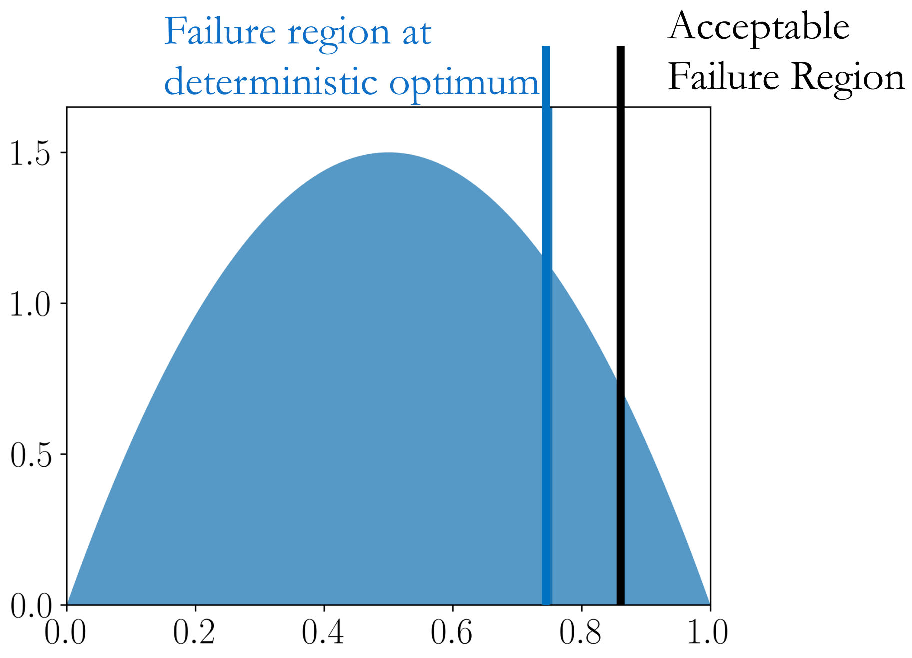
Design under uncertainty

Create design that is robust to uncertainty

$$\min_{\xi\in\Xi} \mathbb{E}\big[\mathcal{L}\big(u(z^\star,\xi)\big)\big]$$
$$s.t.\ \ \mathbb{P}\big[\mathcal{C}\big(u(z,\xi)\big) \leq \delta\big] - \epsilon \leq 0$$

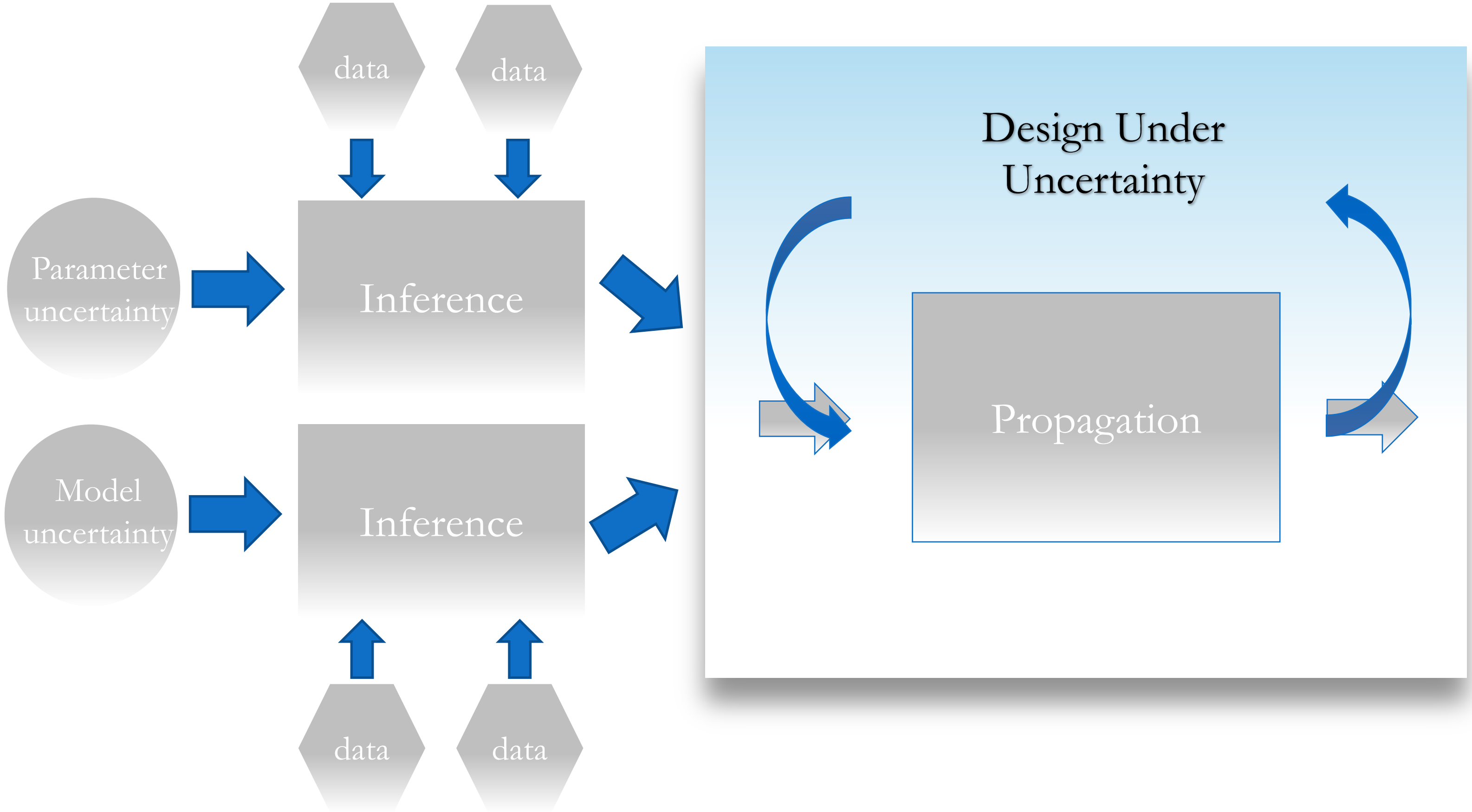


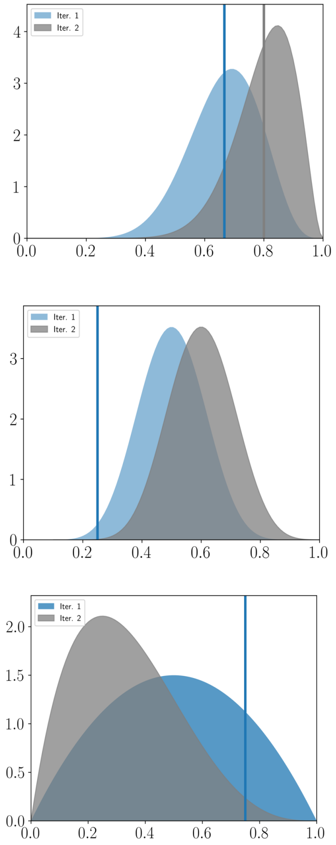
Objective



Constraint

END TO END WORKFLOW



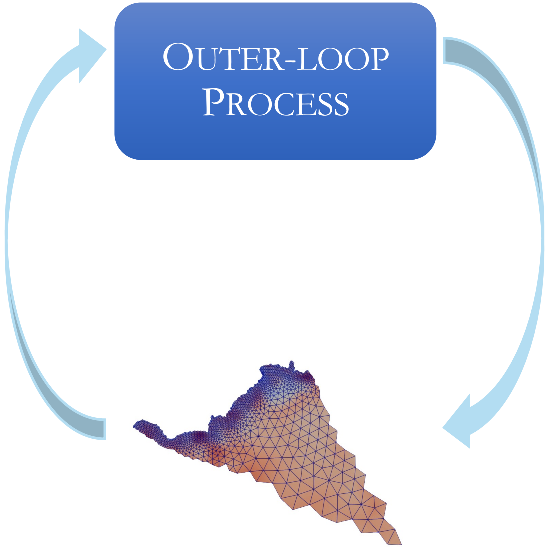


When a model is computational expensive only a limited number of evaluations may be available.

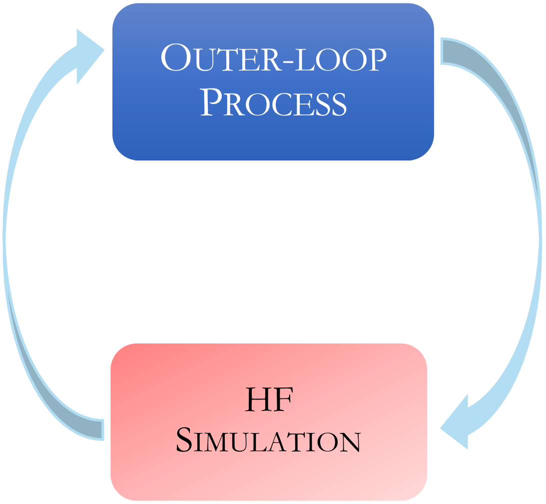
Efficient methods are needed to reduce the computational cost

MULTI-FIDELITY UQ

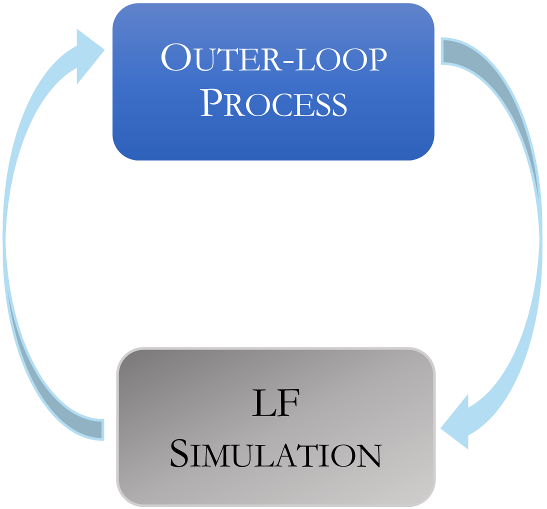
Goal:
Uncertainty
Quantification UQ



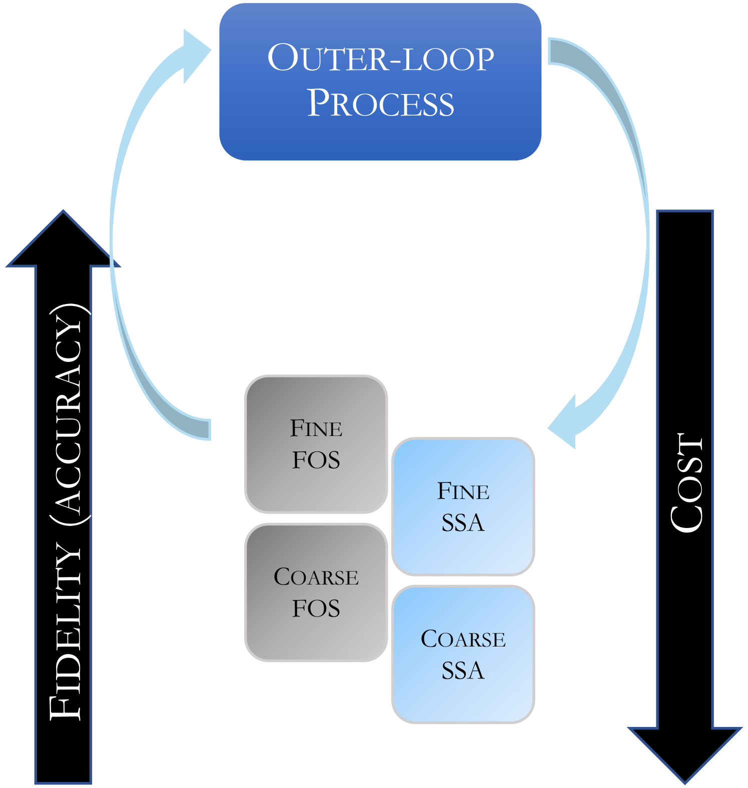
Challenge:
Expensive data intractable



Existing approach:
Use lower-fidelity data



Multi-fidelity solution:
Use data from multiple
models



MF UQ AND MF SURROGATES CAN REDUCE THE COST OF BUILDING SINGLE-FIDELITY BY 10-100X

Numerical discretization impacts Accuracy

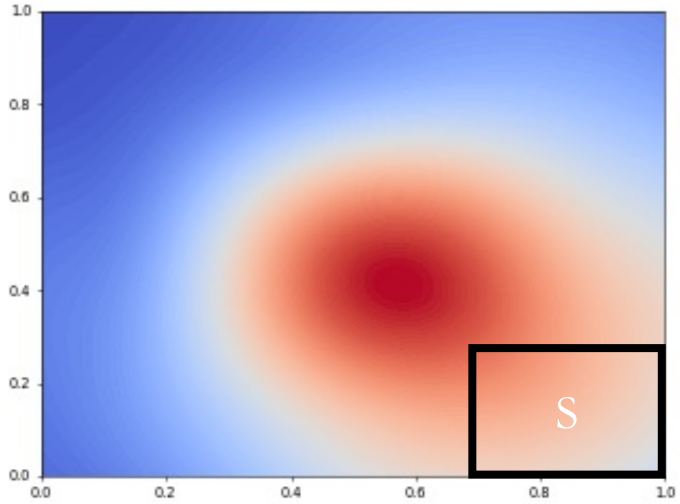
Transient Advection diffusion

$$\frac{\partial u}{\partial t}(x,t,z) = \nabla \cdot [k(x,z)\nabla u(x,t,z)] - \nabla \cdot (vu(x,t,z)) + g(x,t)(x,t,z) \in D \times [0,1] \times \Gamma$$

$$\mathcal{B}(x,t,z) = 0(x,t,z) \in \partial D \times [0,1] \times \Gamma$$

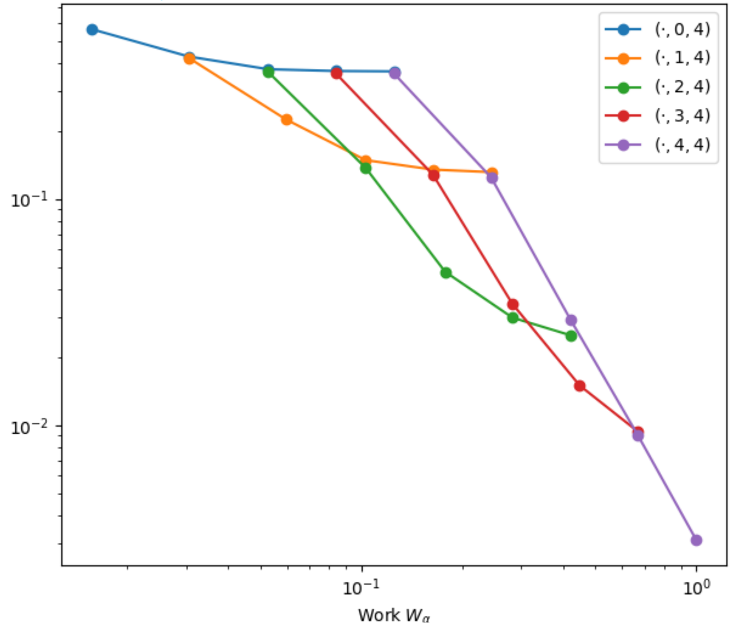
$$u(x,t,z) = u_0(x,z)(x,t,z) \in D \times \{t=0\} \times \Gamma$$

Accuracy of QoI f(z) depends on numerical discretization $f(z) = \int_S u(x,T,z)dx$

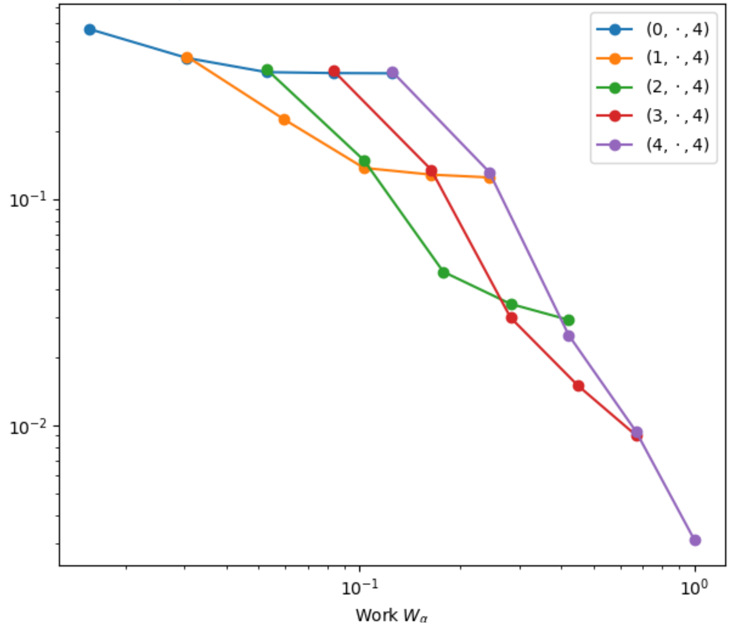


Error in QoI

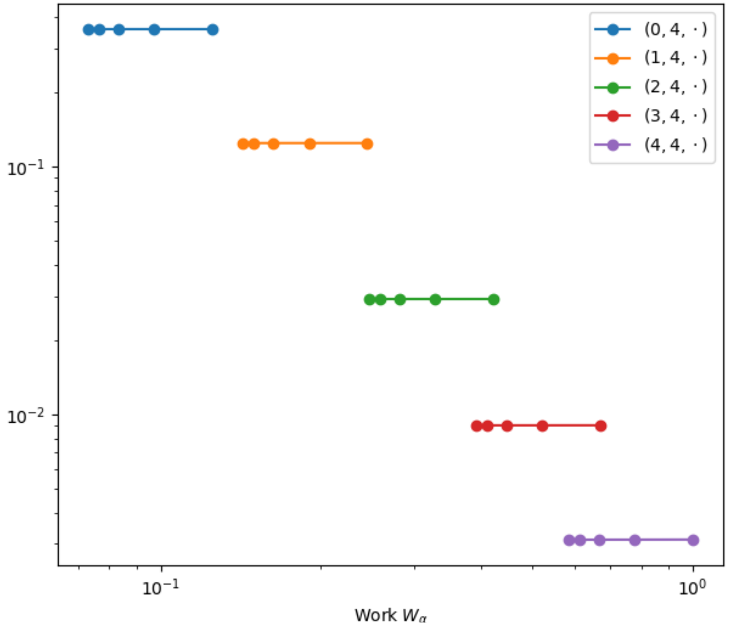
Vary horizontal mesh resolution



Vary vertical mesh resolution

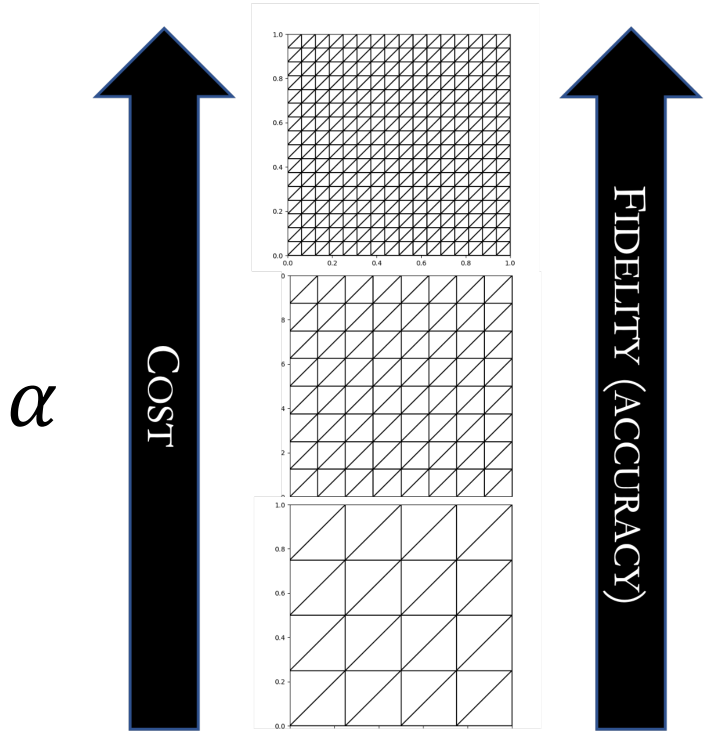


Vary time-step size



MULTI-LEVEL VS MULTI-INDEX

Multi-level methods

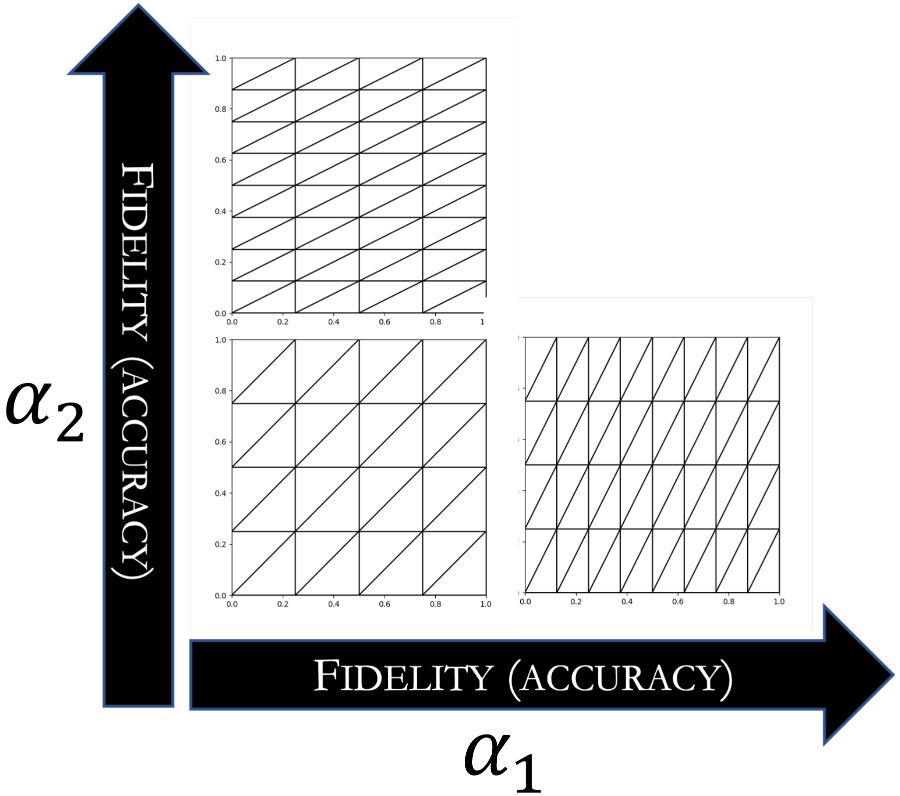


Use a 1D hierarchy of models controlled by a scalar index $\alpha \in \mathbb{N}$

Both assume

$$\|f - f_\alpha\| \leq \|f - f_{\alpha'}\|$$

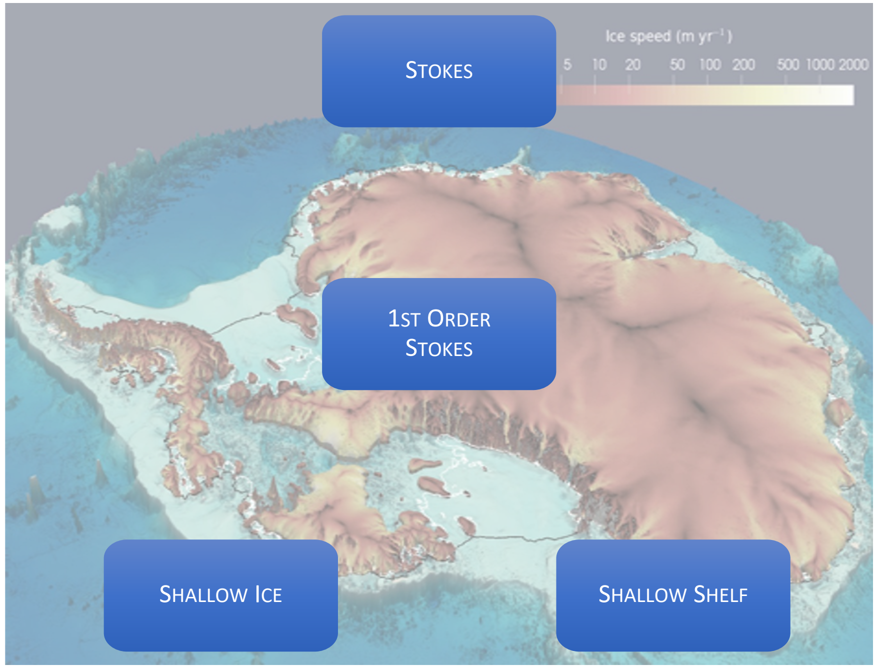
Multi-index methods



Use a multi-dimensional hierarchy controlled by a multi-index $\alpha = [\alpha_1, \alpha_2, \dots]$

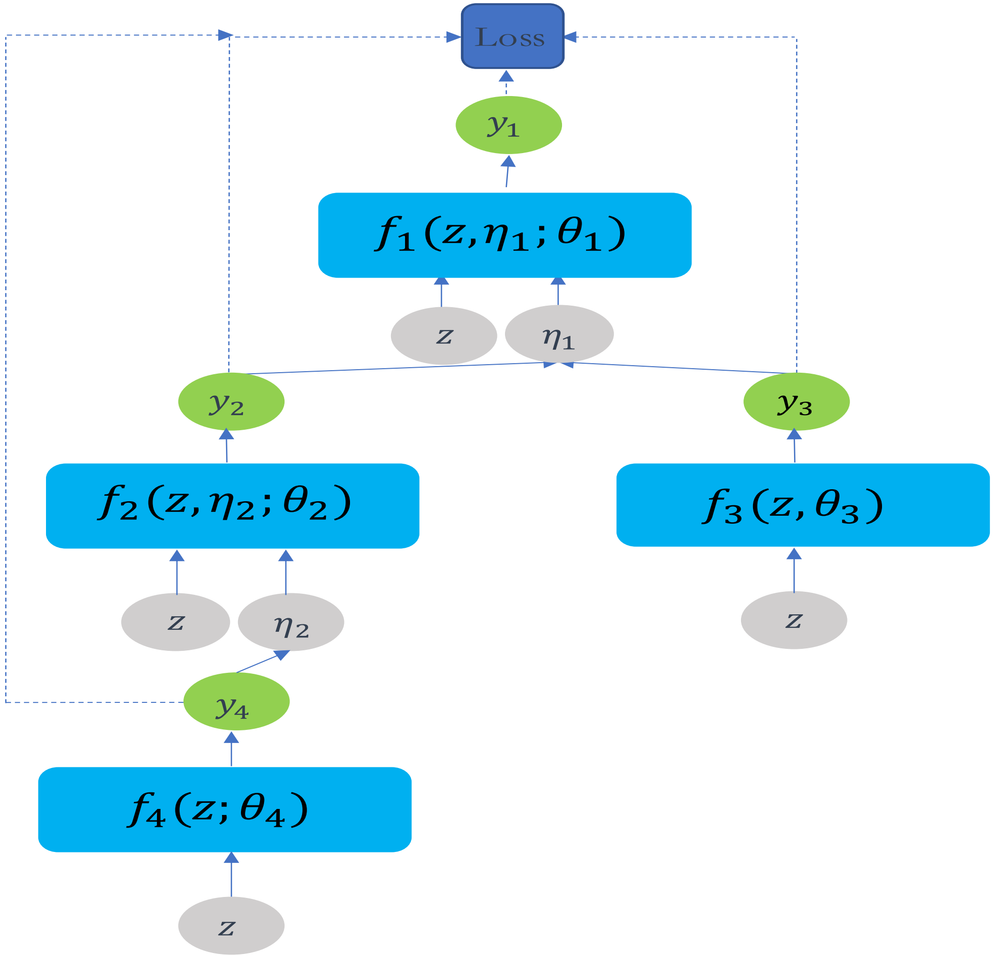
MOVING BEYOND MULTI-INDEX METHODS

Some problems possess an ensemble of models that do not admit a strict hierarchy



Gorodetsky et al. MFNets: Multi-fidelity data-driven networks for bayesian learning and prediction, International Journal for Uncertainty Quantification, 2020.
A. Gorodetsky et al. MFNets: Learning network representations for multifidelity surrogate modeling, 2020.

Methods that can encode general dependencies are now being developed



Multi-fidelity Quadrature

Goal: Compute the expected value using two (or more) models

$$Q^{\mathrm{CV}}_{\boldsymbol{\alpha},N} = Q_{\boldsymbol{\alpha},N} + \eta\left(Q_{\boldsymbol{\kappa},N} - \mu_{\boldsymbol{\kappa}}\right)$$

Minimizing variance ratio to find η

$$\gamma = \frac{\mathbb{V}\left[Q^{\mathrm{CV}}_{\boldsymbol{\alpha},N}\right]}{\mathbb{V}\left[Q_{\boldsymbol{\alpha},N}\right]}$$

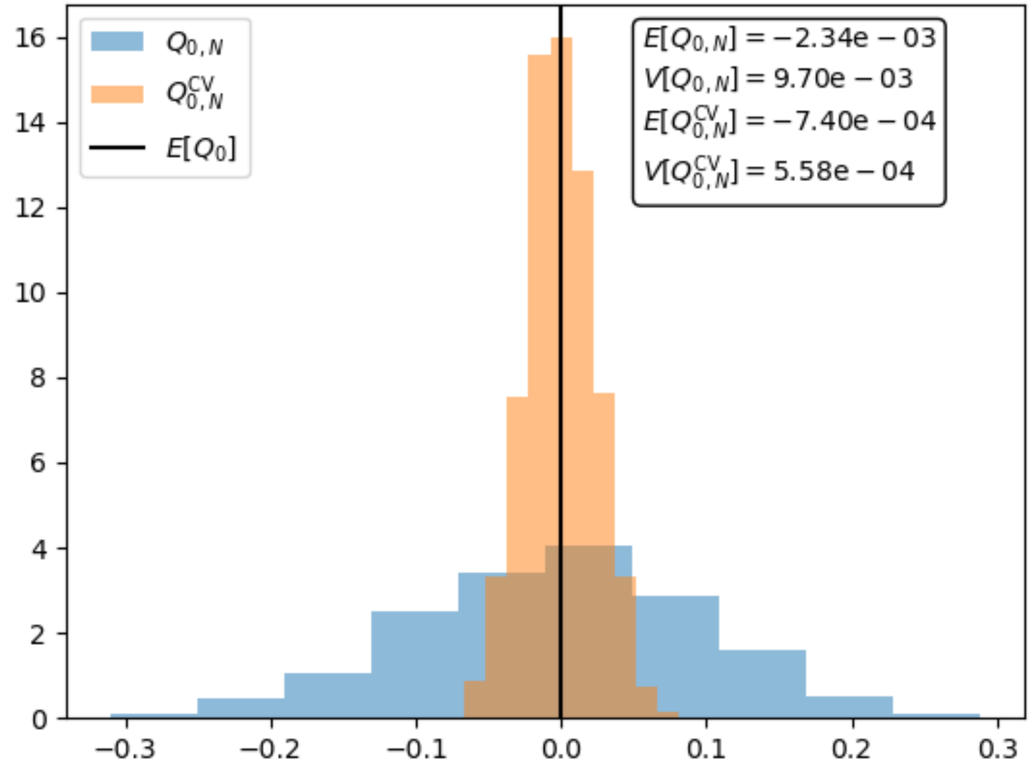
Yields

$$\eta = -\frac{\mathrm{Cov}\left[N^{-1}\sum_i^N f_\alpha\left(z^{(i)}\right), N^{-1}\sum_i^N f_\kappa\left(z^{(i)}\right)\right]}{\mathbb{V}\left[N^{-1}\sum_i^N f_\kappa\left(z^{(i)}\right)\right]} = -\frac{\mathrm{Cov}[f_\alpha, f_\kappa]}{\mathbb{V}[f_\kappa]}$$

Such that

$$\gamma = 1 - \mathbb{C}\mathrm{or}\left[Q_{\boldsymbol{\alpha},N}, Q_{\boldsymbol{\kappa},N}\right]^2$$

If $Q_\alpha = Q$ (unbiased) then CV estimator is also unbiased



But variance of CV estimator is much smaller than HF estimator if models are highly correlated

MULTI-FIDELITY SURROGATES

GOAL

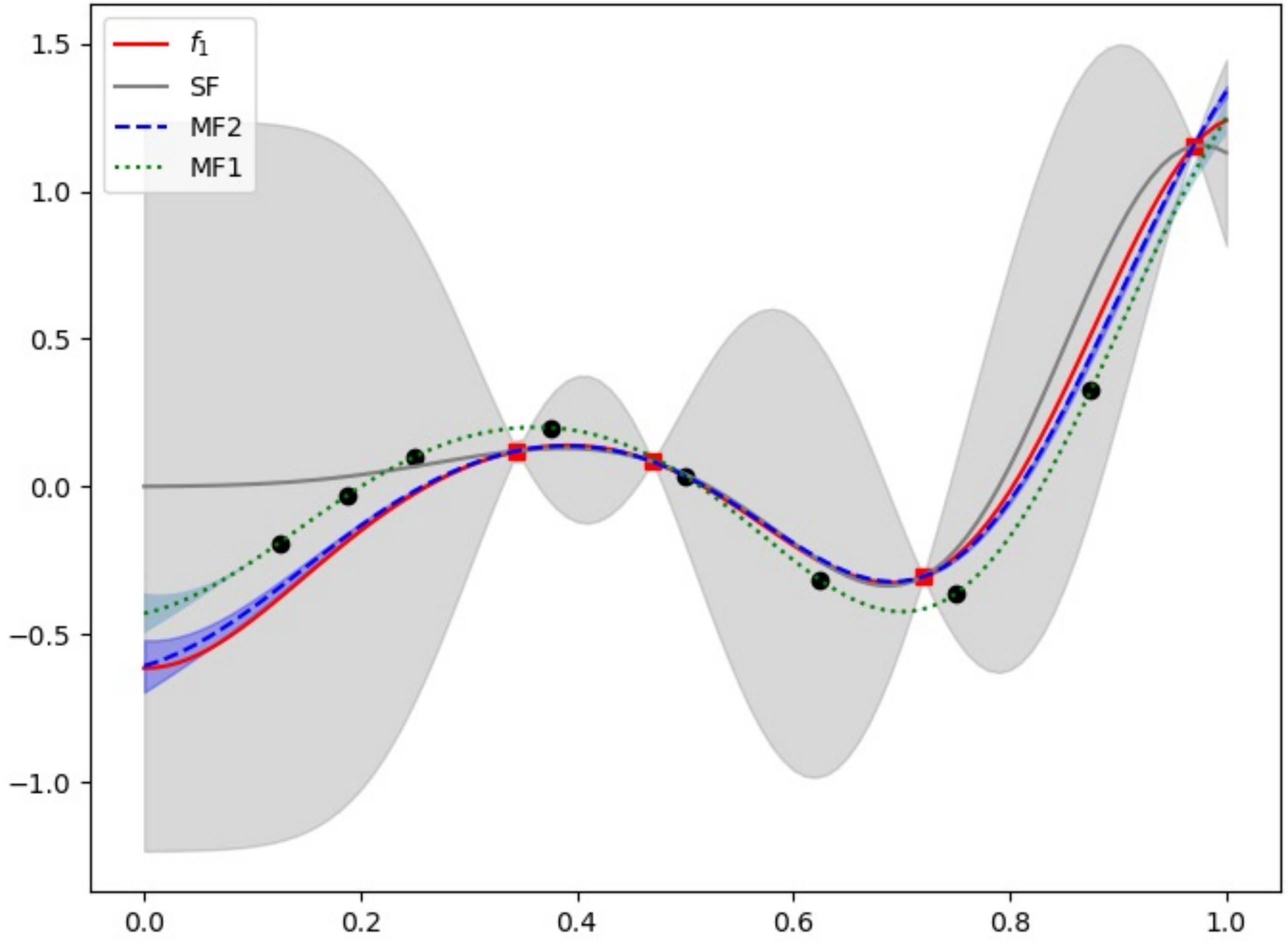
Build a surrogate using data from models of varying fidelity

Set of model fidelities including $\alpha^\star$

$$f_{\mathcal{A},N}(z) \approx f_{\alpha^\star}(z)$$

Num samples per model

Highest fidelity



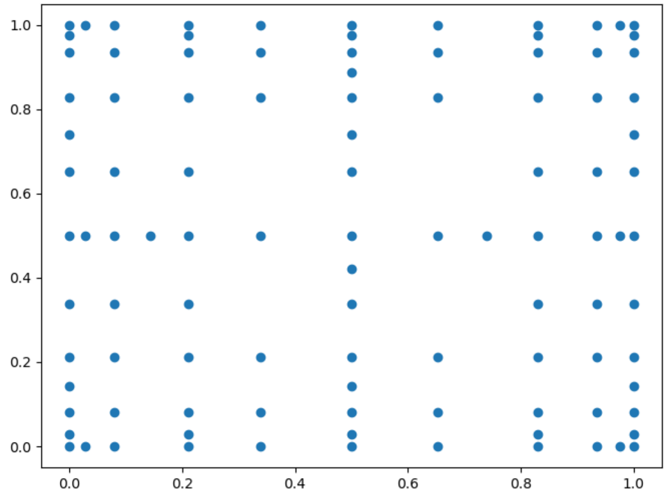
A low-fidelity model can be used to accurately extrapolate away from high-fidelity data

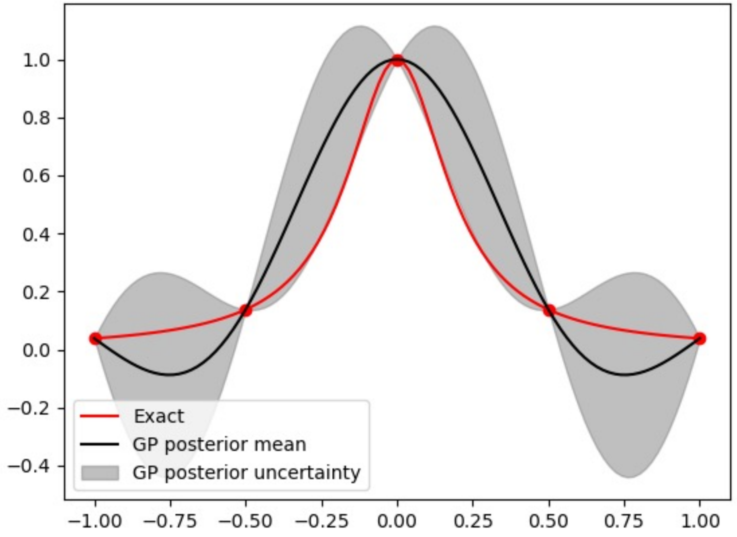
OVERVIEW

1. Multi-level/Multi-index collocation
   - Tensor product interpolation
   - Sparse grid approximation
   - Multi-index collocation

2. Multi-level Gaussian processes
   - Single fidelity Gaussian processes and experimental design
   - Multi-level Gaussian processes and experimental design

3. Alternatives
   - Deep GPs, neural networks, low-rank methods, MFNets



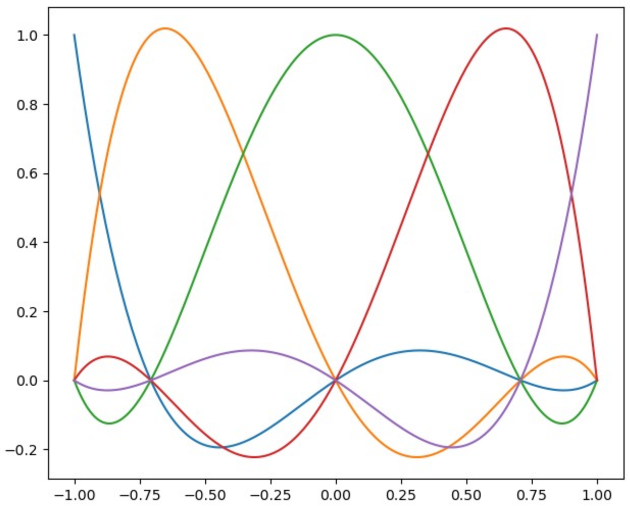


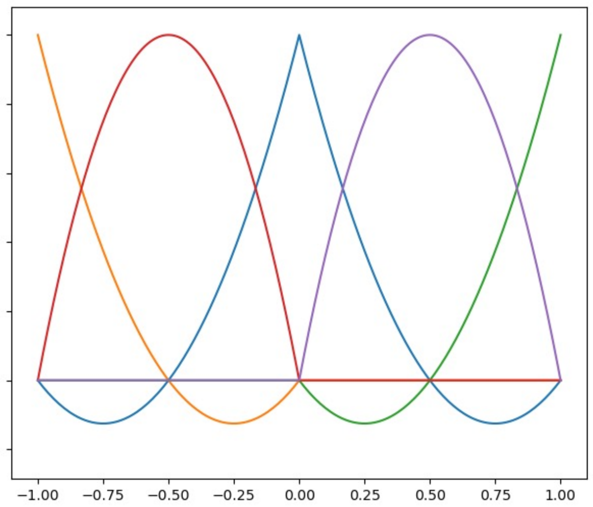
Tensor-product interpolation

Tensor product Lagrange interpolation is based on one dimensional interpolants

$$\phi_{i,j}(z_i) = \prod_{k=1,k\neq j}^{m_{\beta_i}} \frac{z_i - z_i^{(k)}}{z_i^{(j)} - z_i^{(k)}}, \quad i \in [d], \qquad z_i^{(j)} = \cos\left(\frac{(j-1)\pi}{m_{\beta_i}}\right), \qquad j = 1, \ldots, m_{\beta_i}$$

Note other basis functions can be used, e.g. piecewise polynomials.





Lemma 3.1 (Strong convergence [9]).

Let $\nu : \hat{\Xi} \to \mathbb{R}$ *and* $\omega : \Xi \to \mathbb{R}$ *denote two densities which satisfy*

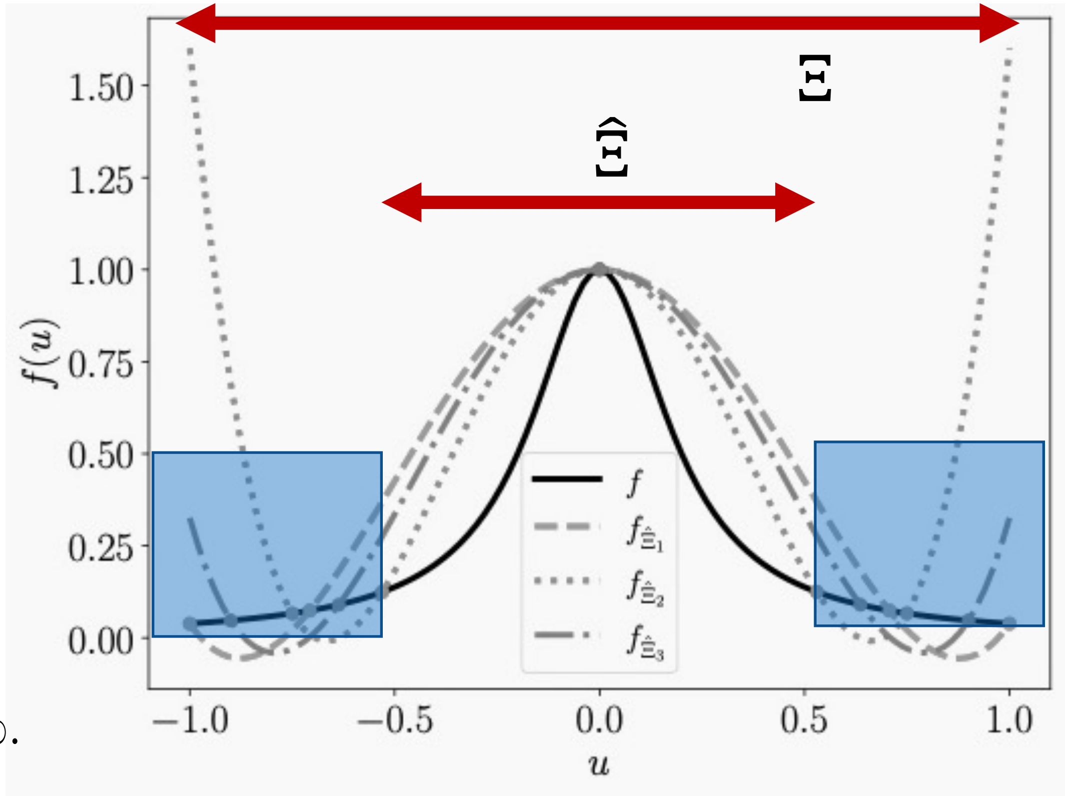
$$\delta = 1 - \int_{\Xi \cap \hat{\Xi}} \omega(\boldsymbol{u}) \mathrm{d}\boldsymbol{u}$$

Given an approximation f_ν *of* f *with approximation error* ϵ, *i.e.,*

$$\epsilon := \|f - f_\nu\|_{L^p_\nu(\Xi)}, \quad p \geq 1,$$

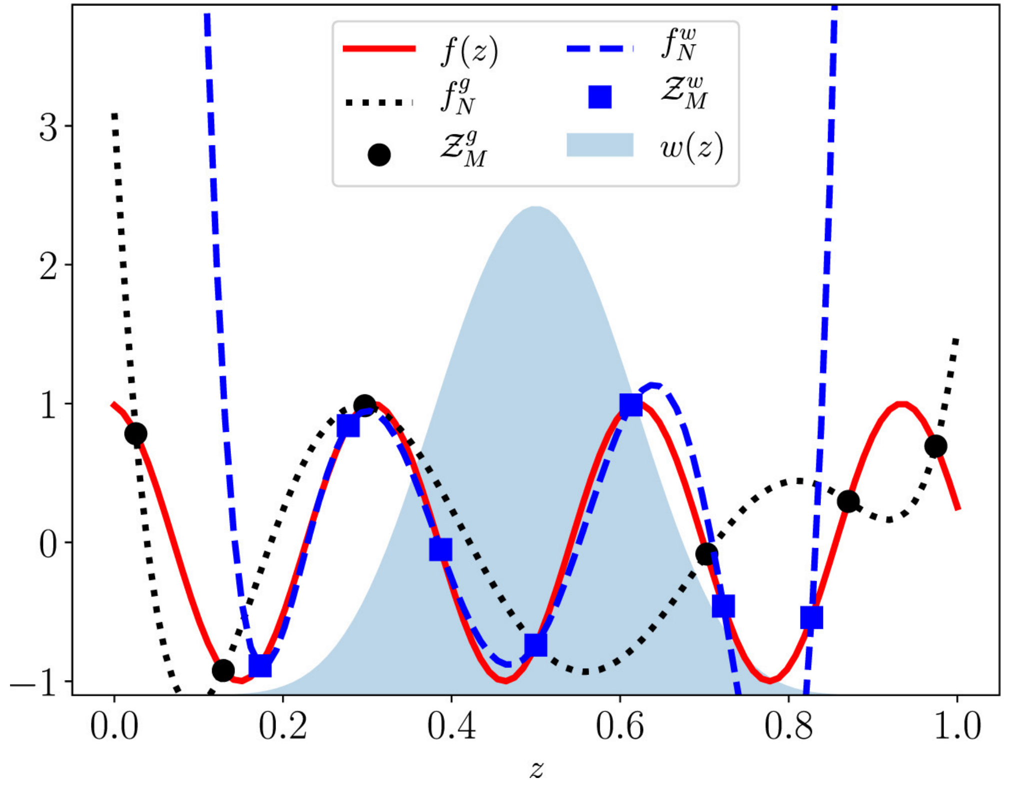
then, if f *is bounded with* $C_f = \|f\|_{L^\infty(\Xi)}$, *it holds that*

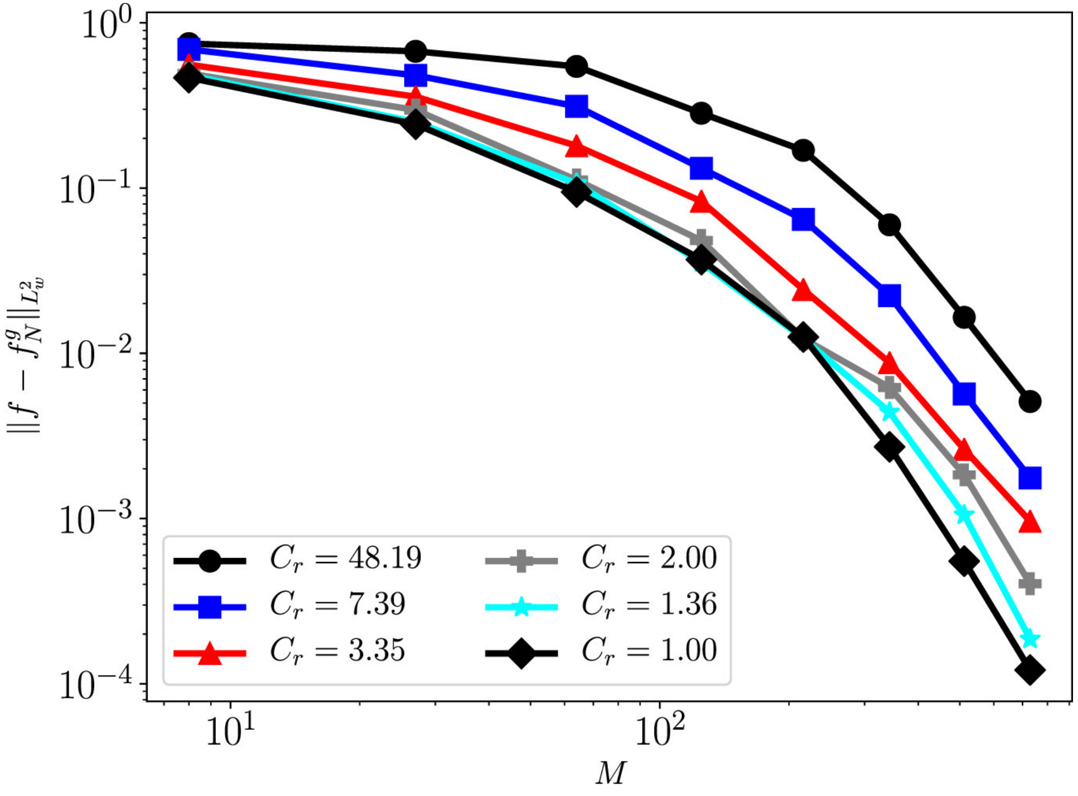
$$\|f - f_\nu\|_{L^p_\omega(\Xi)} \leq C_r^{1/p} \epsilon + C_f \delta^{1/p}, \quad \textit{provided} \quad C_r := \max_{\boldsymbol{u} \in \Xi \cup \hat{\Xi}} \frac{\omega(\boldsymbol{u})}{\nu(\boldsymbol{u})} < \infty.$$



Interpolants for an oscillatory function parameterized by a Beta(10,10) random variable

The constant C_r of convergence significantly impacts error





Tensor-product interpolation

The tensor product interpolant is

$$\hat{f}_{\boldsymbol{\alpha},\boldsymbol{\beta}}(\mathbf{z}) = \sum_{\boldsymbol{j} \leq \boldsymbol{\beta}} \hat{f}_{\boldsymbol{\alpha}}(\mathbf{z}^{(\boldsymbol{j})}) \prod_{i \in [d]} \phi_{i,j_i}(z_i).$$

It requires evaluating the model on the tensor product grid

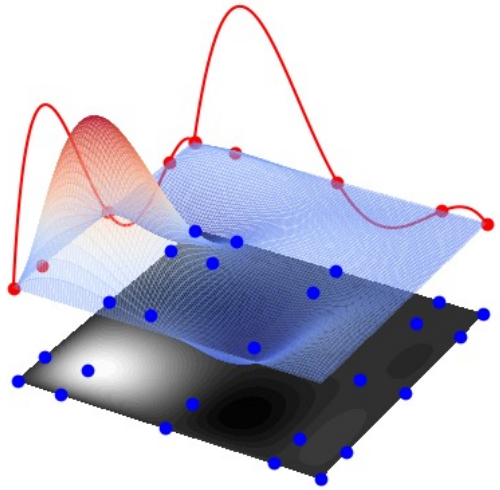
$$\mathcal{Z}_{\boldsymbol{\beta}} = \bigotimes_{i=1}^{d} \mathcal{Z}^{i}_{\beta_i} = \left[\mathbf{z}^{(1)} \quad \cdots \quad \mathbf{z}^{(M_{\boldsymbol{\beta}})}\right] \in \mathbb{R}^{d \times M_{\boldsymbol{\beta}}}$$

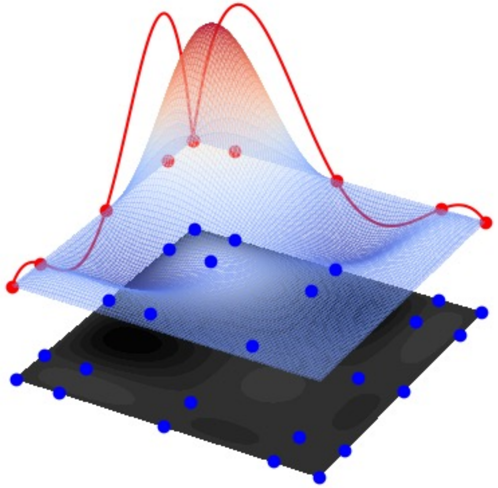
With number of points

$$M_{\boldsymbol{\beta}} = \prod_{i \in [d]} m_{\beta_i}$$

And model evaluations

$$\mathcal{F}_{\boldsymbol{\alpha},\boldsymbol{\beta}} = \hat{f}_{\boldsymbol{\alpha}}(\mathcal{Z}_{\boldsymbol{\beta}}) = \left[\hat{f}_{\boldsymbol{\alpha}}(\mathbf{z}^{(1)}) \quad \cdots \quad \hat{f}_{\boldsymbol{\alpha}}(\mathbf{z}^{(M_{\boldsymbol{\beta}})})\right]^{T} \in \mathbb{R}^{M_{\boldsymbol{\beta}} \times q},$$

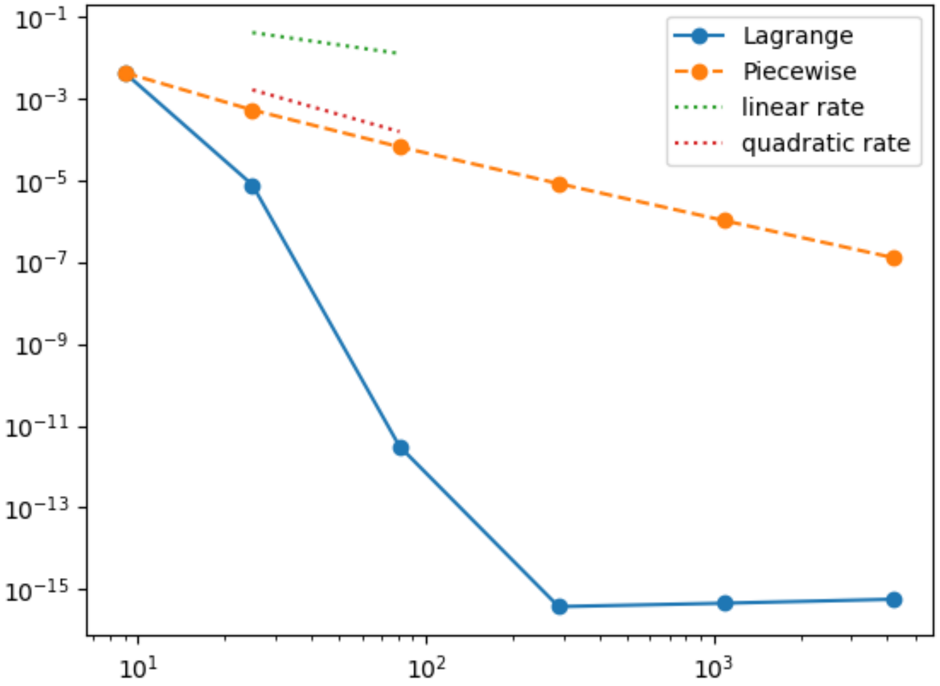




SMOOTHNESS MATTERS

The performance of Lagrange-based tensor products depends on the smoothness s and the dimension of the target function

$$\|f_\alpha - f_{\alpha,\beta}\|_{L^\infty(\Gamma)} \leq C_{d,s} N_\beta^{-s/d}$$



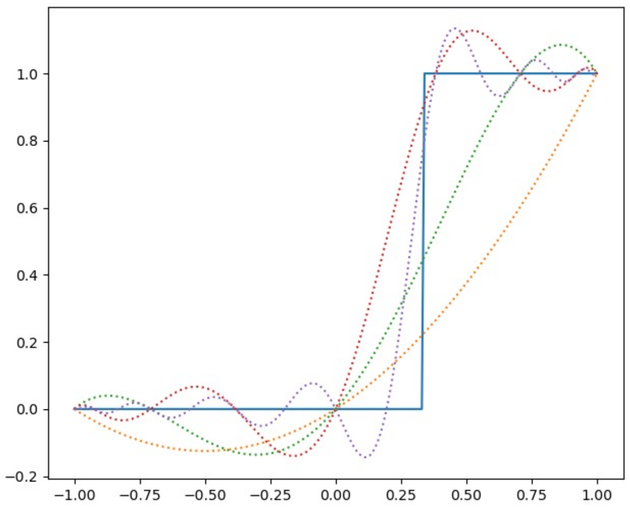
$$f(z) = \cos\left(2\pi w_1 + \sum_{d=1}^{D} c_d z_d\right)$$

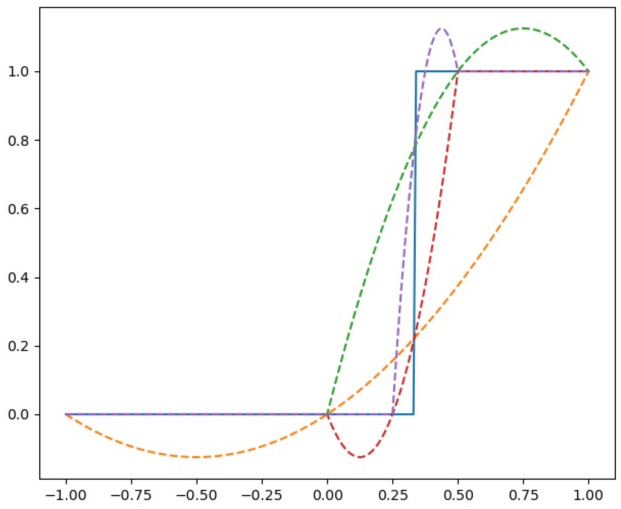
Exponential convergence is obtained for analytic functions

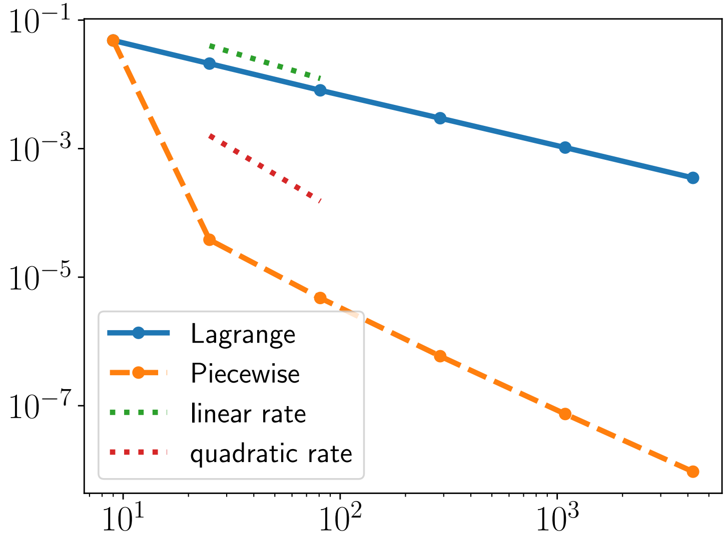
SMOOTHNESS MATTERS

Lagrange polynomials introduce Gibbs like phenomena when approximating discontinuous functions

Only linear convergence is obtained for functions with only integrable first order derivatives







$$f(z) = \exp\left(-\sum_{d=1}^{D} c_d |z_d - w_d|\right)$$

QUADRATURE USING THE SURROGATE

The error in the Monte Carlo estimate of the mean using the surrogate satisfies

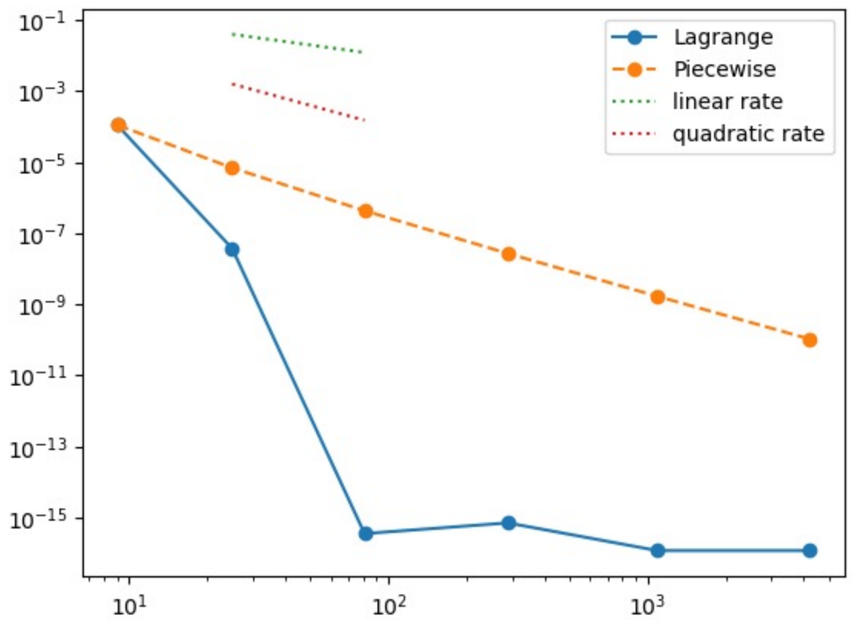
$$\mathbb{E}\left[(Q_\alpha - \mathbb{E}[Q])^2\right] = N^{-1}\mathbb{V}[Q_\alpha] + (\mathbb{E}[Q_\alpha] - \mathbb{E}[Q])^2$$
$$\leq N^{-1}\mathbb{V}[Q_\alpha] + C_{d,r}N_\beta^{-s/d}$$

Unlike the expensive model, the first term can be made very small because the surrogate is cheap to evaluate

However the mean of tensor product interpolants can be computed exactly.

$$\mu_\beta = \int_\Gamma \sum_{\boldsymbol{j}\leq\beta} f_\alpha(z^{(\boldsymbol{j})}) \prod_{i=1}^{d} \phi_{i,j_i}(z_i) w(z)\, dz = \sum_{\boldsymbol{j}\leq\beta} f_\alpha(z^{(\boldsymbol{j})}) v_{\boldsymbol{j}}.$$

$$v_{\boldsymbol{j}} = \prod_{i=1}^{d} \int_{\Gamma_i} \phi_{i,j_i}(z_i)\, dw(z_i),$$



$$f(z) = \cos\left(2\pi w_1 + \sum_{d=1}^{D} c_d z_d\right)$$

SPARSE GRID APPROXIMATION

The number of tensor product points grows exponentially with dimension (curse of dimensionality)

Sparse grids can be used to exploit function smoothness to mitigate the curse of dimensionality

Index set controlling accuracy

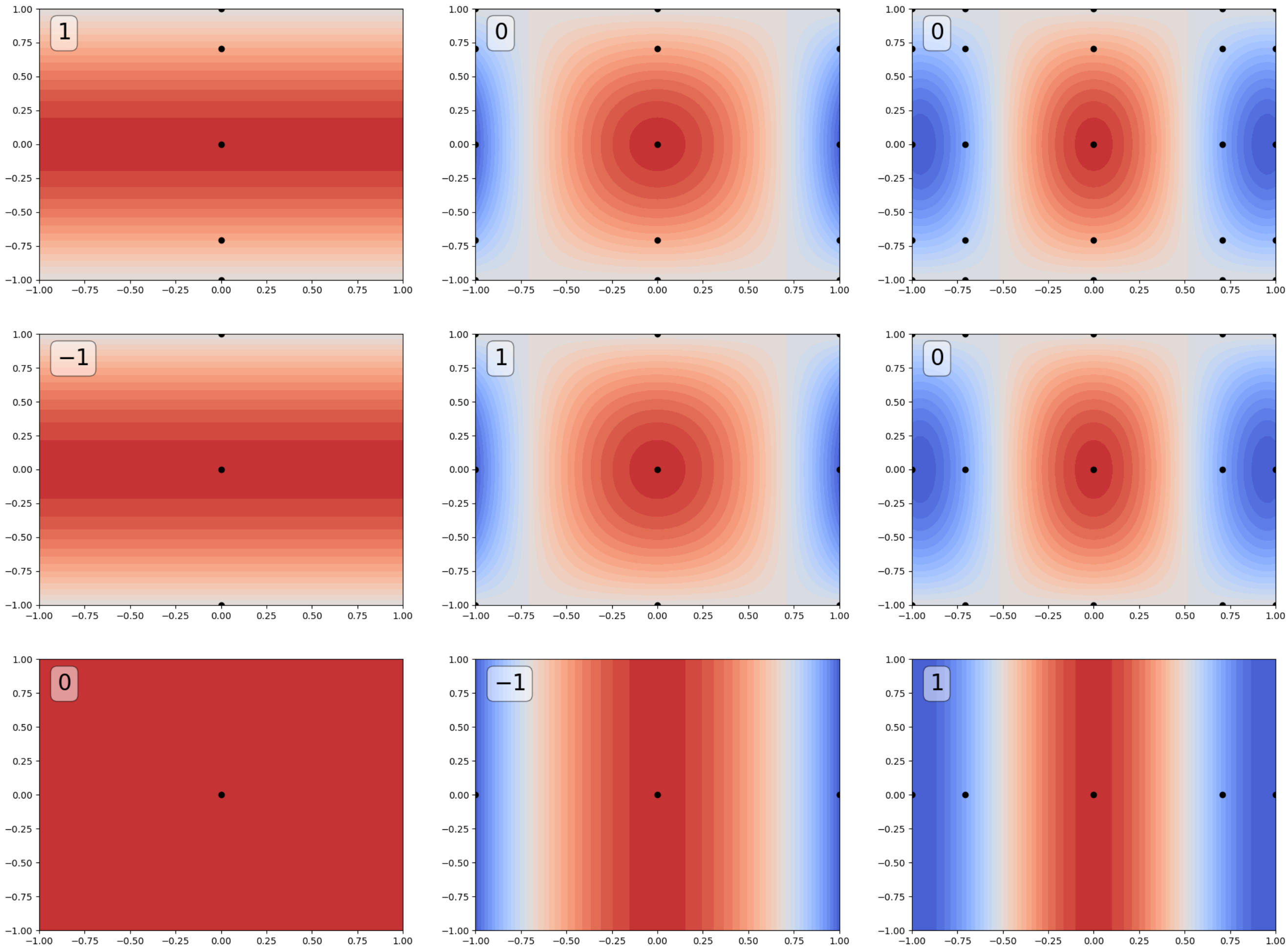
TP approx

$$f_{\alpha,\mathcal{I}}(z) = \sum_{\beta \in \mathcal{I}} c_\beta f_{\alpha,\beta}(z),$$

numerical model discretization

Tensor product resolution

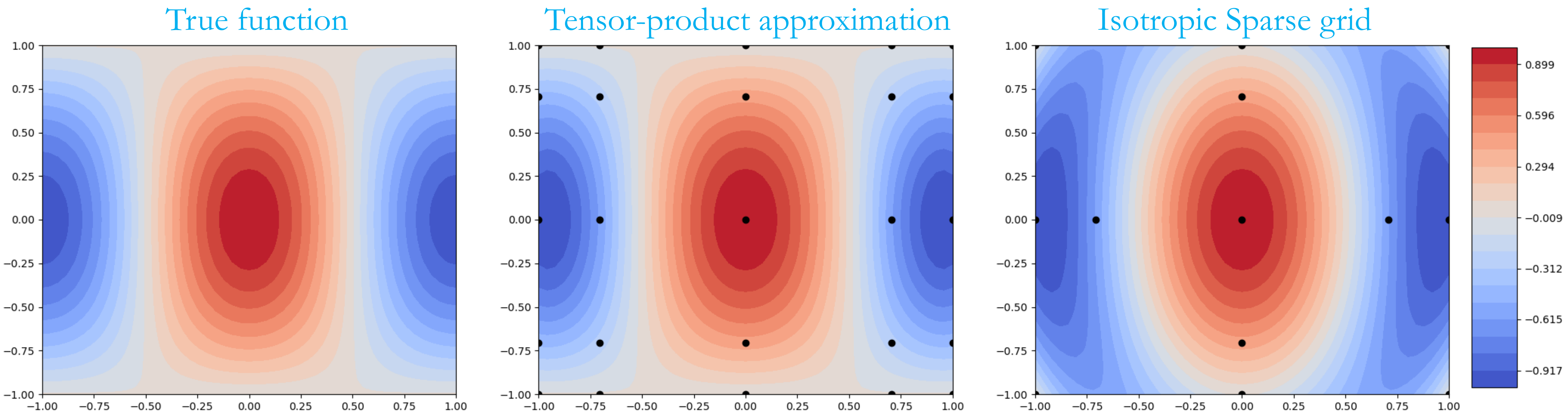
Isotropic sparse grid



ISOTROPIC SPARSE GRID

An isotropic sparse grid uses

$$\mathcal{I}(l) = \{\beta \mid (\max(0, l-1) \leq \|\beta\|_1 \leq l + D - 2\}, \quad l \geq 0 \qquad c_\beta = (-1)^{l-|\beta|_1}\binom{D-1}{l-|\beta|_1}.$$



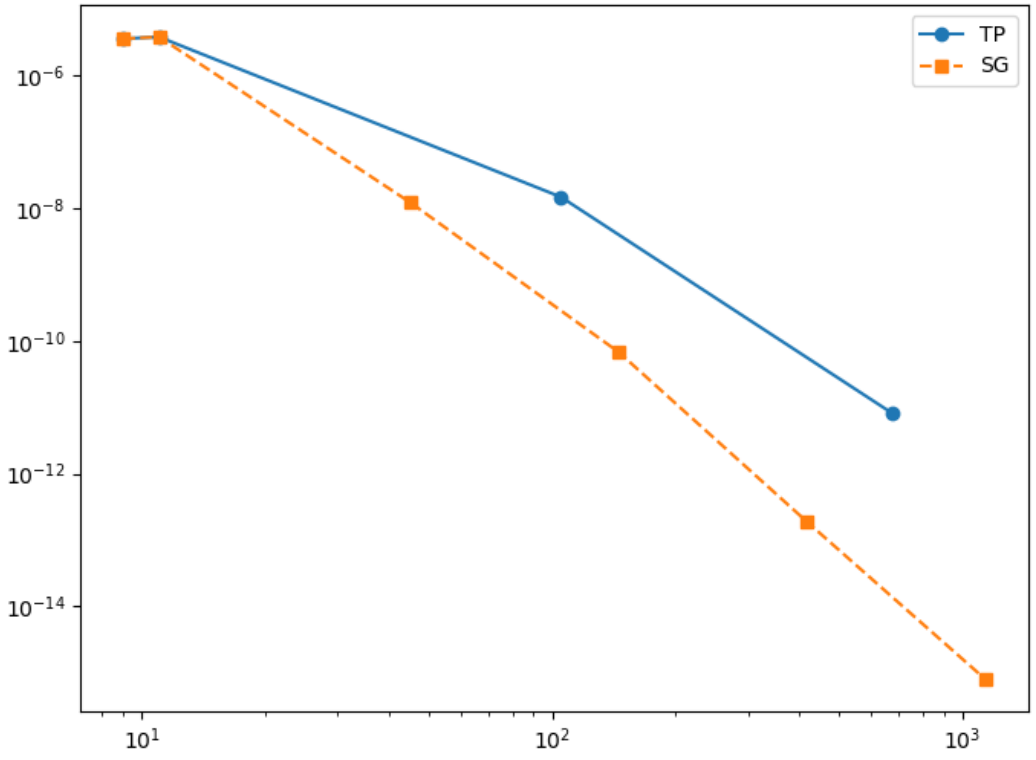
V. Barthelmann, E. Novak and K. Ritter. High dimensional polynomial interpolation on sparse grid. Advances in Computational Mathematics (2000).
H. Bungartz and M. Griebel. Sparse grids. Acta Numerica (2004).

SPARSE GRID CONVERGENCE

$$f(z) = \exp\left(-\frac{\sum_{d=1}^{4}(0.5(z_d+1)-0.5)^2}{20}\right)$$

Tensor product error grows exponentially with dimension for fixed smoothness r

$$\|f - f_{\mathcal{I}(l)}\|_{L^\infty} \leq K_{D,r} M_l^{-r/D}$$



Isotropic sparse error grids is less strongly dependent on dimension

$$\|f - f_{\mathcal{I}(l)}\|_{L^\infty} \leq C_{D,r} M_{\mathcal{I}(l)}^{-r} (\log M_{\mathcal{I}(l)})^{(r+2)(D-1)+1}$$

Isotropic grids treat all dimensions equally, but for many models some dimensions are more important than others

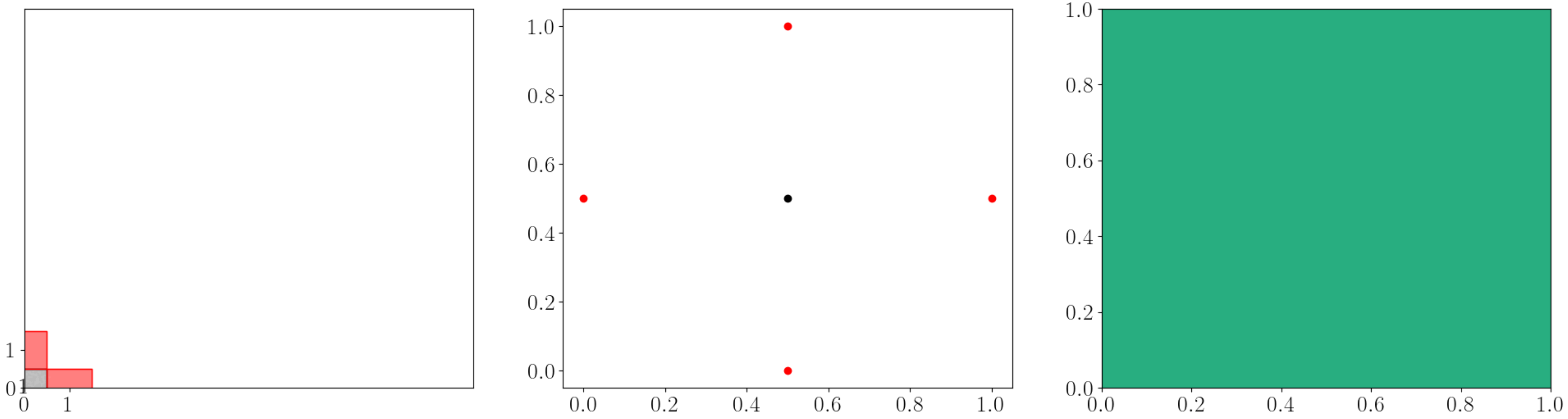
ADAPTIVE SPARSE GRIDS

Finding the optimal index set can be posed as binary knapsack problem

$$\max \sum_{\beta} \Delta E_\beta \delta_\beta \text{ such that } \sum_{\beta} \Delta W_\beta \delta_\beta \le W_{\max},$$

$$\Delta E_\beta = \|f_{\alpha,\mathcal{I}\cup\beta} - f_{\alpha,\mathcal{I}}\| \qquad \Delta W_\beta = \|W_{\alpha,\mathcal{I}\cup\beta} - W_{\alpha,\mathcal{I}}\|$$

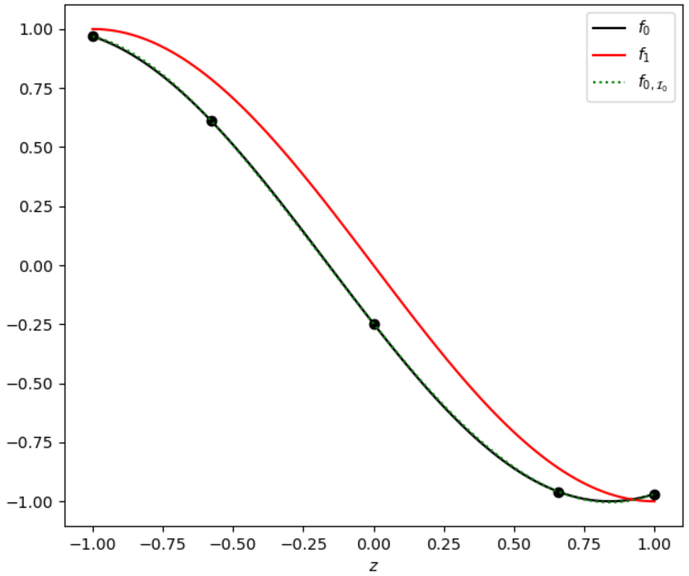
A greedy algorithm can be used to find an approximate solution



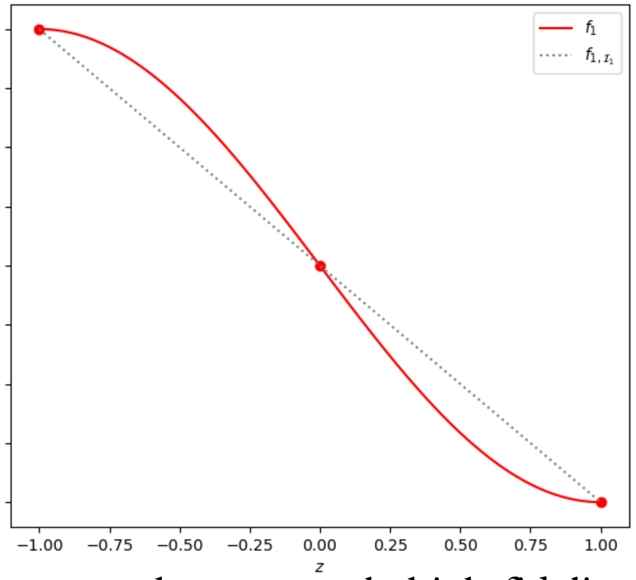
T. Gerstner and M. Griebel. Dimension-adaptive tensor-product quadrature. Computing (2003).

Multi-fidelity approximation: an observation

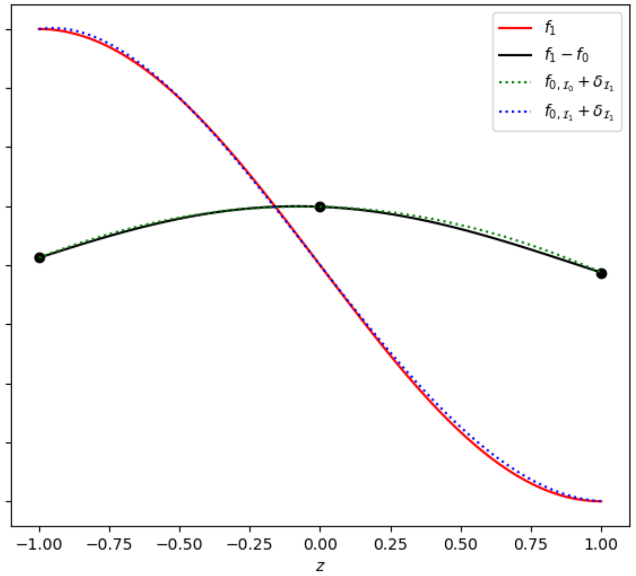
The discrepancy between model fidelities is often "easier" to approximate than the high-fidelity function



Surrogate that uses only low fidelity model f_{1,N_1} has small stochastic error but has large deterministic error



Surrogate that uses only high fidelity model f_{0,N_0} has large stochastic error but has no deterministic error



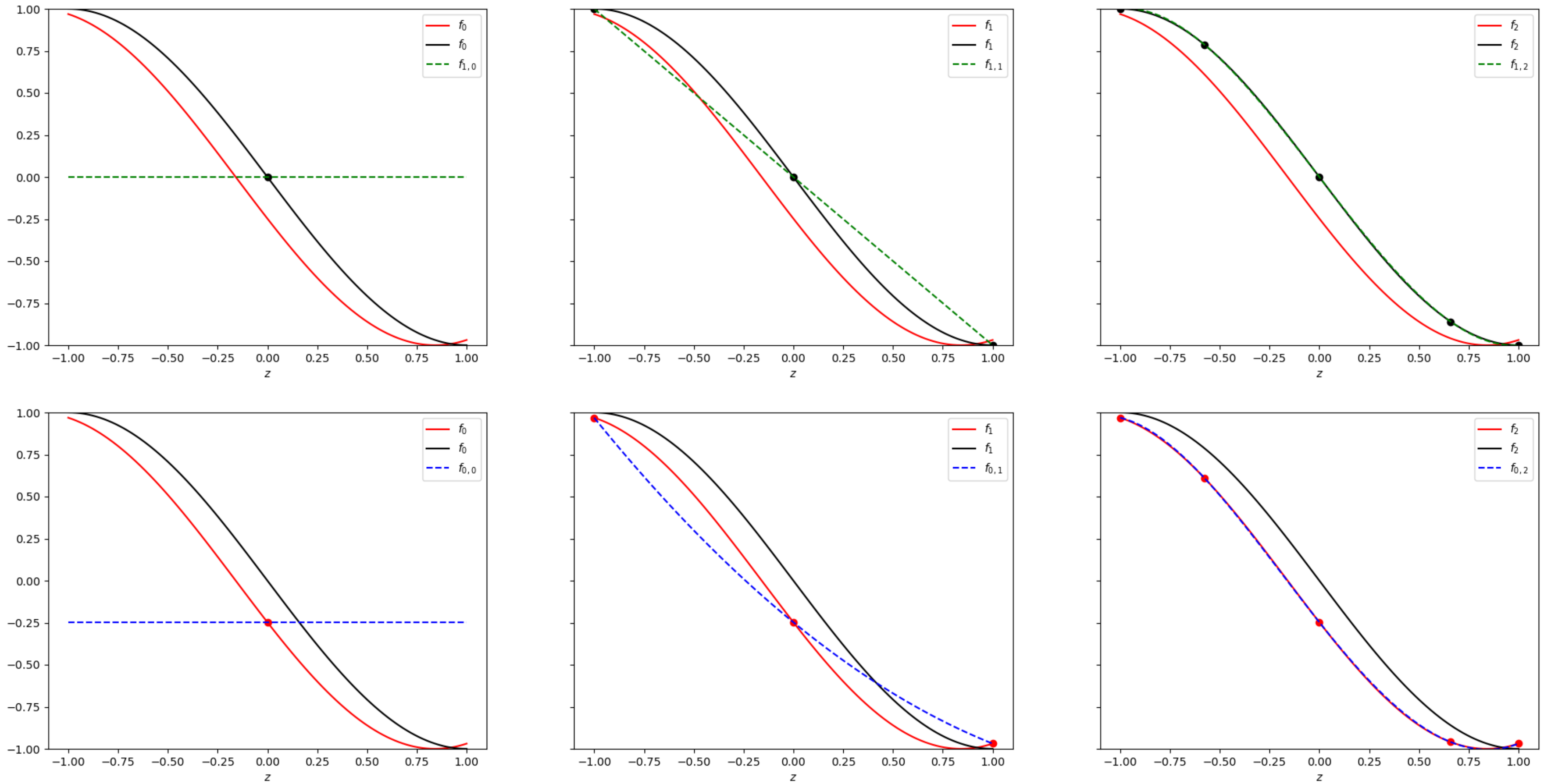
MF Surrogate has small stochastic error and has no deterministic error

Multi-index Collocation

$$f_{\mathcal{I}}(z) = \sum_{[\alpha,\beta]\in\mathcal{I}} c_{[\alpha,\beta]} f_{\alpha,\beta}(z)$$

Like sparse grids MISC is a linear combination of tensor product interpolants

However, now additional indices are used to increase the fidelity of data being used

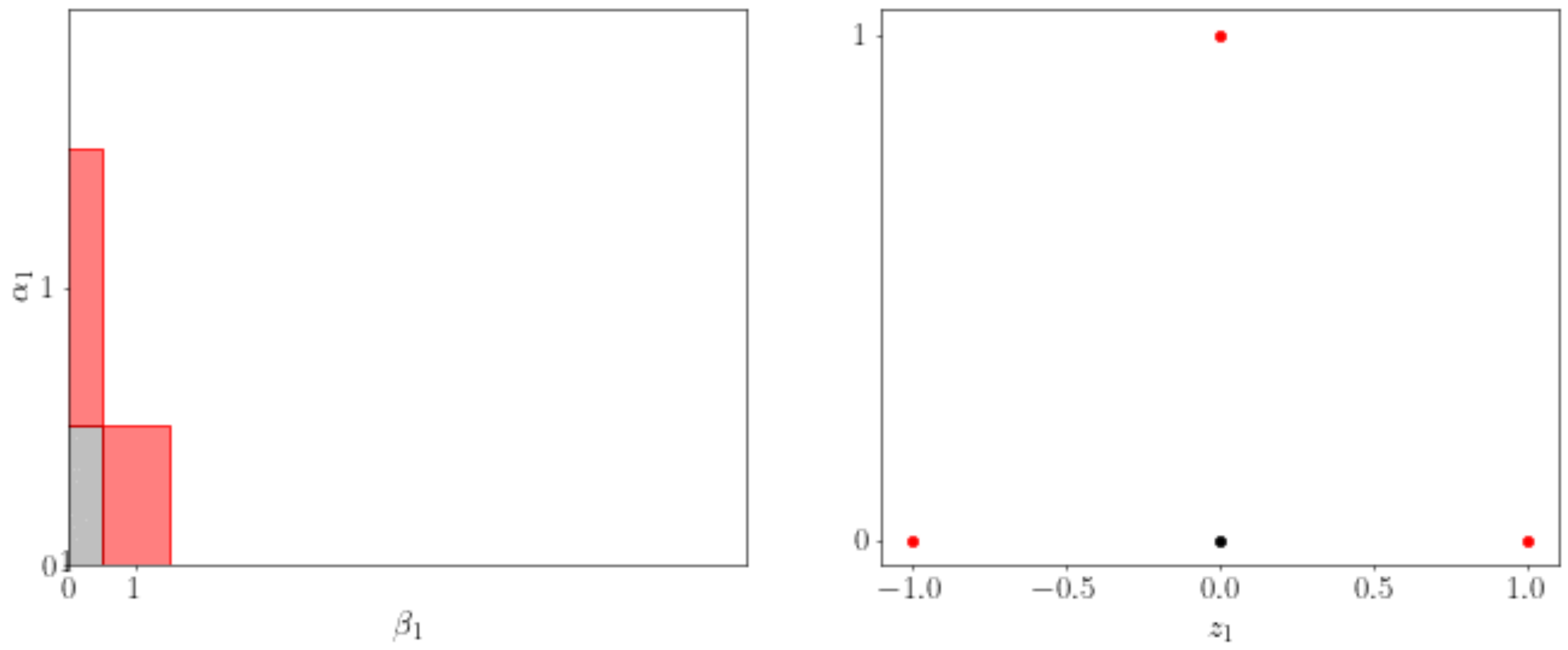


Haji-Ali, F. Nobile, L. Tamellini, and R. Tempone. Multi-index stochastic collocation for random pdes. Computer Methods in Applied Mechanics and Engineering, 306:95 – 122, 2016.

A. Teckentrup, P. Jantsch, C. Webster, and M. Gunzburger. A multilevel stochastic collocation method for partial differential equations with random input data. SIAM/ASA Journal on Uncertainty Quantification, 3(1):1046-1074, 2015.

ADAPTIVE MULTI-INDEX COLLOCATION

The sparse grid adaptation algorithm can be modified for use with multi-fidelity models



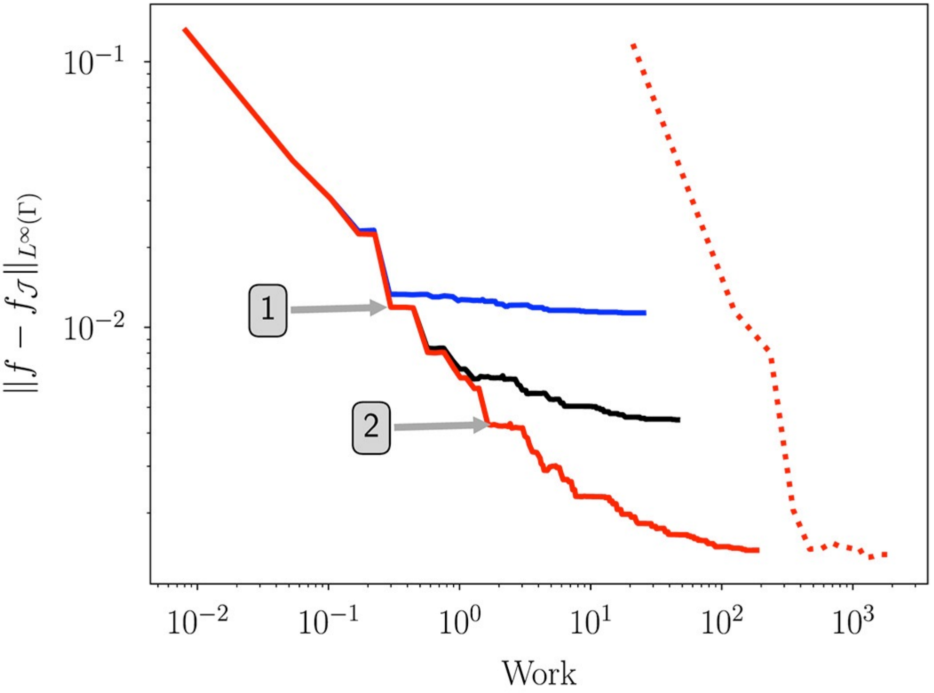
J.D. Jakeman, M.S. Eldred, G. Geraci, and A. Gorodetsky. Adaptive Multi-index Collocation for Uncertainty Quantification and Sensitivity Analysis. International Journal for Numerical Methods in Engineering (2019).

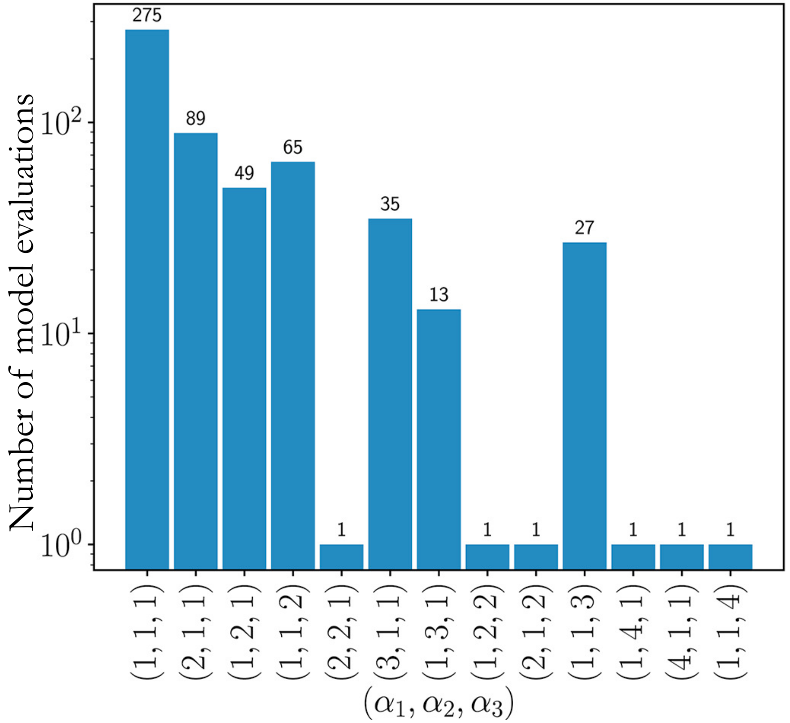
Advection Diffusion Model

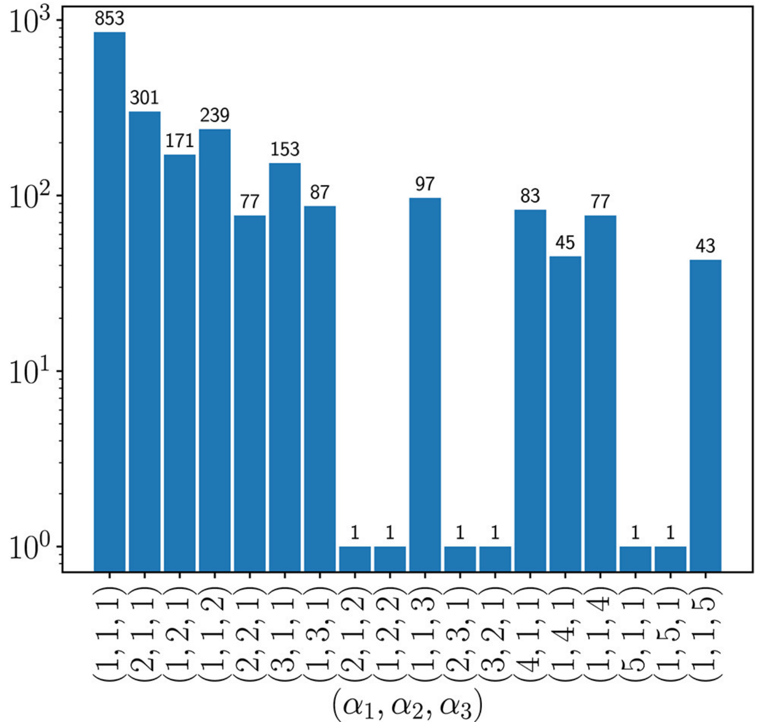
MISC reduces the computational cost of building a surrogate relative to a single fidelity sparse grid for a 3D hierarchy (x-refinement, y refinement and time refinement)

Work is allocated to each model according to cost relative to the improvement in predictive accuracy.

Low fidelity meshes are used early on and higher-fidelity meshes when requesting higher accuracy







GAUSSIAN PROCESSES

Gaussian processes are a distribution over a class of functions

Mean function

$$f(\cdot) \mid \theta \sim \mathcal{N}\left(m(\cdot), C(\cdot,\cdot;\theta) + \epsilon^2 I\right)$$

Hyper-parameters

Noise variance

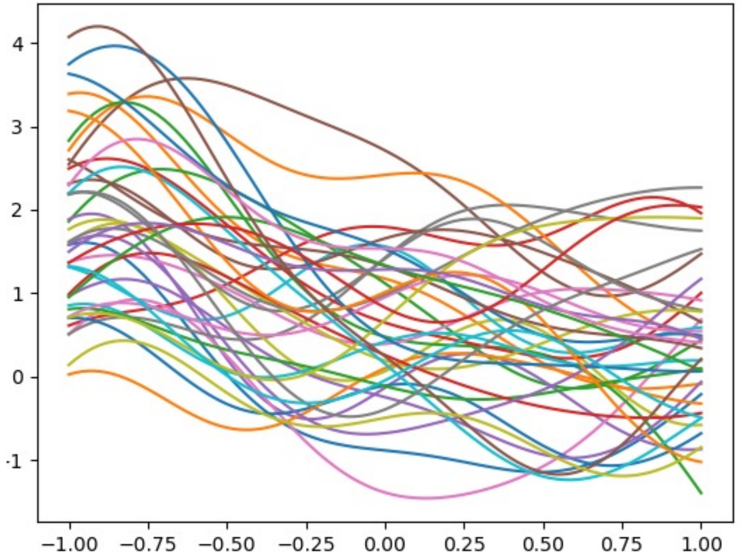
The kernel C should be tailored to the smoothness of the function being approximated. The Matern kernel is a flexible choice

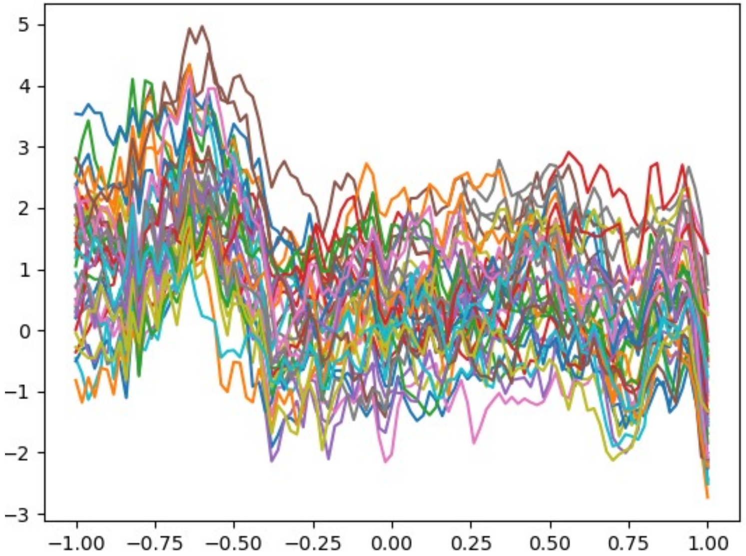
$$C_\nu(z, z^\star; \theta) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu} d(z, z^\star; \ell)}{\ell} \right)^\nu K_\nu \left(\frac{\sqrt{2\nu} d(z, z^\star; \ell)}{\ell} \right).$$

Controls smoothness

Kernel variance. Controls magnitude of function

Length scale. Controls frequency of realizations





Gaussian processes

The posterior distribution of the Gaussian processes conditional on training data $\left(z^{(i)}, y^{(i)} = f(z^{(i)})\right)$ is

$$f(\cdot) \mid \theta, y \sim \mathcal{N}\left(m^\star(\cdot), C^\star(\cdot,\cdot;\theta) + \epsilon^2 I\right)$$

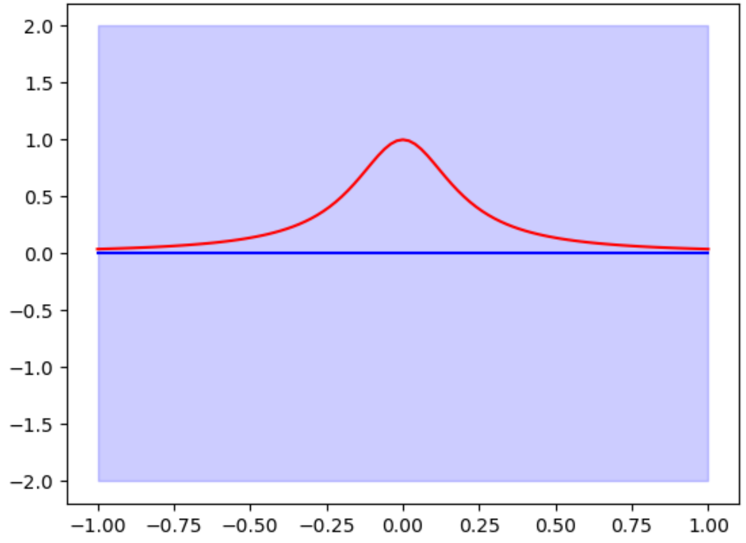
With posterior mean and covariance

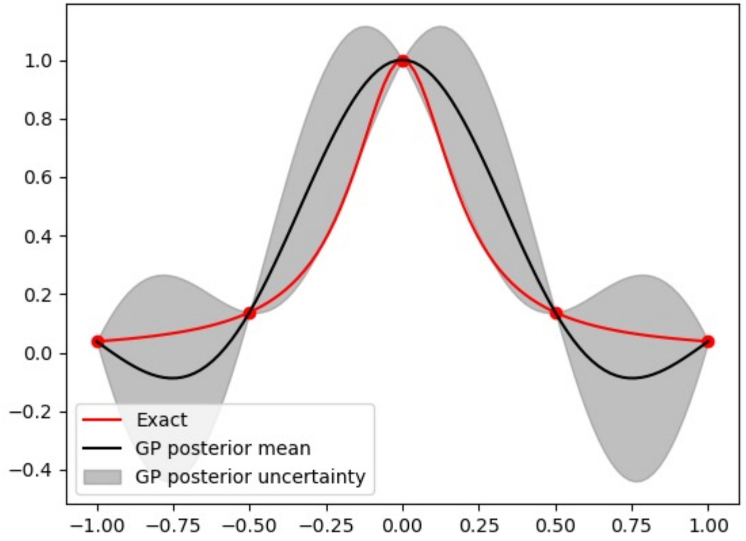
$$m^\star(z) = t(z)^\top A^{-1} y \qquad C^\star(z,z') = C(z,z') - t(z)^\top A^{-1} t(z')$$

where

$$A_{ij} = C(z^{(i)}, z^{(j)}) \text{ for } i,j = 1,\ldots,M$$

$$t(z) = [C(z,z^{(1)}),\ldots,C(z,z^{(N)})]^\top$$





C.E. Rasmussen and C. WIlliams. Gaussian Processes for Machine Learning. MIT Press (2006)

Experimental Design

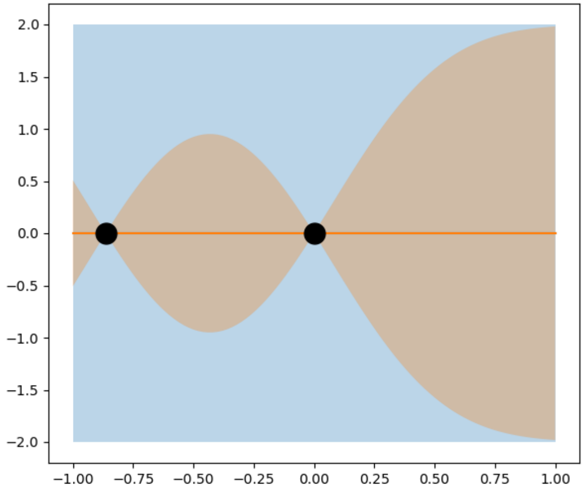
Not all training data reduce variance equally.

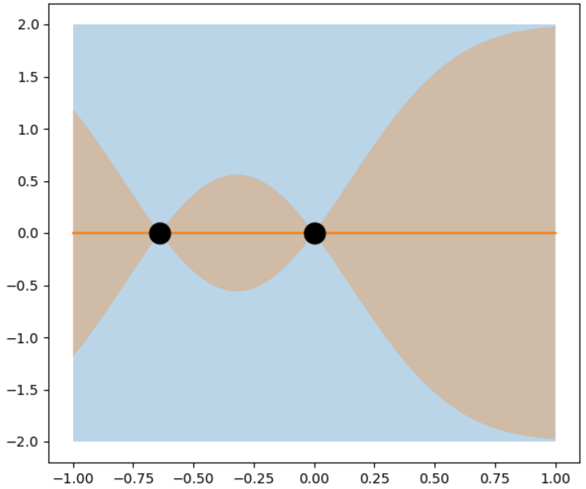
Experimental design can be used to reduce variance systematically.

Integrated variance (IVAR) designs minimize the posterior distribution of the GP with respect to the distribution of the inputs z

$$\mathcal{Z}^{\dagger} = \underset{\mathcal{Z}\subset\Omega\subset\Gamma,|\mathcal{Z}|=M}{\operatorname{argmin}} \int_{\Gamma} C^{\star}(z, z \mid \mathcal{Z})\rho(z)dz$$

Note the designs do not depend on the data and can be computed a priori





J. Sacks, W.J. Welch, T.J.Mitchell, H.P. Wynn Designs and analysis of computer experiments (with discussion). Statistical Science, 4:409-435 (1989)

IVAR Experimental Design

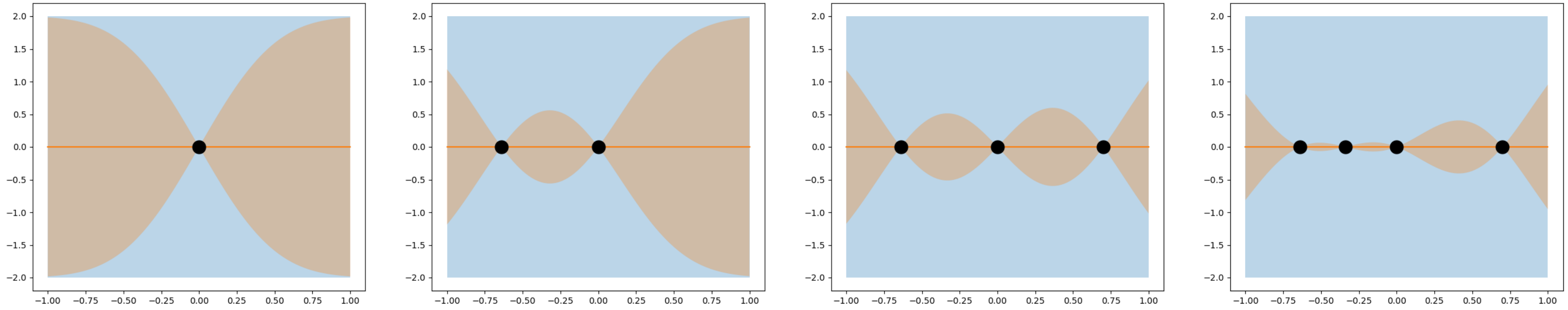
The IVAR objective simplifies to

$$\int_\Gamma C^\star(z, z \mid \mathcal{Z})\rho(z)dz = 1 - \text{Trace}\left[A_{\mathcal{Z}} P_{\mathcal{Z}}\right] \qquad P_{\mathcal{Z}} = \int_\Gamma A_{\mathcal{Z}\cup\{z\}} A^\top_{\mathcal{Z}\cup\{z\}} \rho(z)dz$$

Which is typically solved greedily such that

$$z_{N+1} = \underset{z' \in \mathcal{Z}_{\text{cand}}}{\text{argmin}} \; \text{Trace}\left[A_{\mathcal{Z}_N \cup\{z'\}} P_{\mathcal{Z}_N \cup\{z'\}}\right].$$

When ρ and C are separable then P can be computed using 1D quadrature



Multilevel Gaussian Processes

Multilevel GPs assume

$$f_m(z) = \rho_{m-1} f_{m-1}(z) + \delta_m(z)$$

For $\mathcal{Z} = (\mathcal{Z}_1, \dots, \mathcal{Z}_M)$, the posterior mean and covariance again satisfy

$$m^\star(z) = t(z)^\top C(\mathcal{Z}, \mathcal{Z})^{-1} y$$

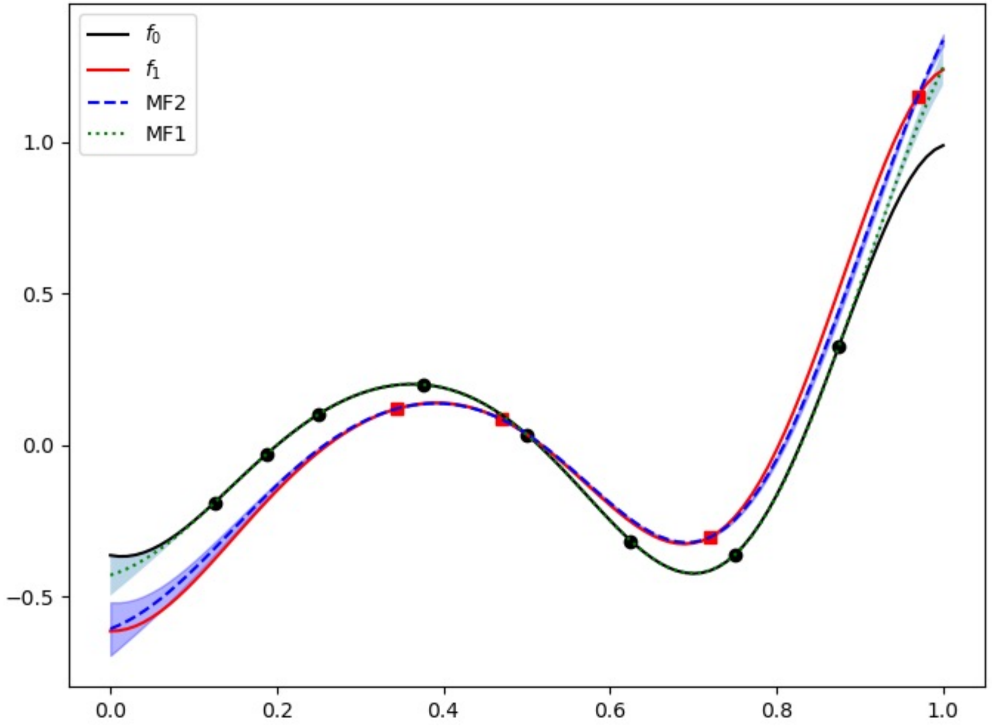
$$C^\star(z, z') = C(z, z') - t(z)^\top C(\mathcal{Z}, \mathcal{Z})^{-1} t(z')$$

Where for two models

$$C(\mathcal{Z}, \mathcal{Z}) = \begin{bmatrix} \mathbb{C}\text{ov}\left[f_1(\mathcal{Z}_1), f_1(\mathcal{Z}_1)\right] & \mathbb{C}\text{ov}\left[f_1(\mathcal{Z}_1), f_2(\mathcal{Z}_2)\right] \\ \mathbb{C}\text{ov}\left[f_2(\mathcal{Z}_2), f_1(\mathcal{Z}_1)\right] & \mathbb{C}\text{ov}\left[f_2(\mathcal{Z}_2), f_2(\mathcal{Z}_2)\right] \end{bmatrix}$$

$$f_1(z) = \frac{1}{5}\left((3y-1)^2 + 1\right)\sin((10y-2))$$

$$f_2(z) = \frac{9}{10} f_1(z) + \left(\frac{2y-1}{4}\right)$$



M. C. Kennedy and A. O'Hagan. Predicting the Output from a Complex Computer Code When Fast Approximations Are Available. Biometrika, 87(1), 1-13, 2000.

MULTILEVEL KERNELS

For two models $\mathcal{Z} = [\mathcal{Z}_1, \mathcal{Z}_2]$

$$C(\mathcal{Z}, \mathcal{Z}) = \begin{bmatrix} C_1(\mathcal{Z}_1, \mathcal{Z}_1) & \rho_1 C_1(\mathcal{Z}_1, \mathcal{Z}_2) \\ \rho_1 C_1(\mathcal{Z}_2, \mathcal{Z}_1) & \rho_1^2 C_1(\mathcal{Z}_2, \mathcal{Z}_2) + C_2(\mathcal{Z}_2, \mathcal{Z}_2) \end{bmatrix}$$

The low-fidelity covariance is

$$\mathbb{C}\text{ov}\left[f_1(\mathcal{Z}_1), f_1(\mathcal{Z}_1)\right] = \mathbb{C}\text{ov}\left[\delta_1(\mathcal{Z}_1), \delta_1(\mathcal{Z}_1)\right] = C_1(\mathcal{Z}_1, \mathcal{Z}_1)$$

The high-fidelity covariance is

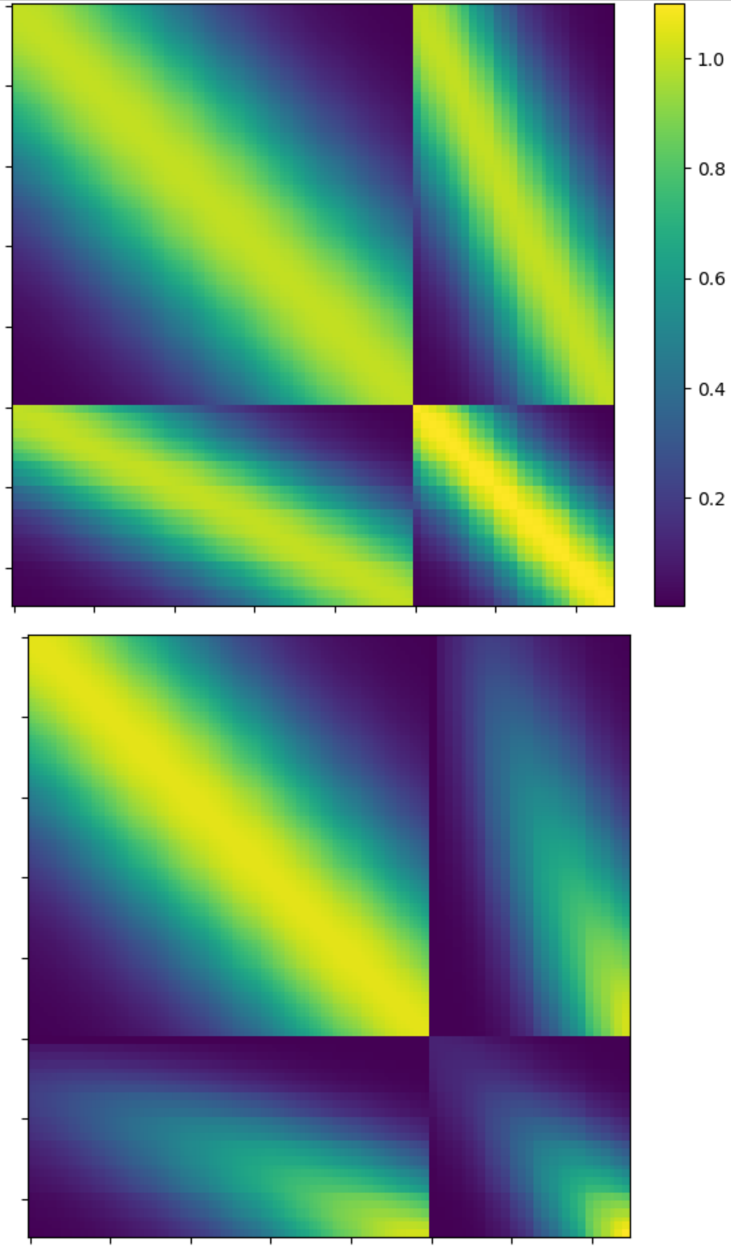
$$\begin{aligned} \mathbb{C}\text{ov}\left[f_2(\mathcal{Z}_2), f_2(\mathcal{Z}_2)\right] &= \mathbb{C}\text{ov}\left[\rho_1 f_1(\mathcal{Z}_2) + \delta_2(\mathcal{Z}_2), \rho_1 f_1(\mathcal{Z}_2) + \delta_2(\mathcal{Z}_2)\right] \\ &= \mathbb{C}\text{ov}\left[\rho_1 \delta_2(\mathcal{Z}_2) + \delta_2(\mathcal{Z}_2), \rho_1 \delta_1(\mathcal{Z}_2) + \delta_2(\mathcal{Z}_2)\right] \\ &= \mathbb{C}\text{ov}\left[\rho_1 \delta_2(\mathcal{Z}_1), \rho_1 \delta_1(\mathcal{Z}_2)\right] + \mathbb{C}\text{ov}\left[\delta_2(\mathcal{Z}_2), \delta_2(\mathcal{Z}_2)\right] \\ &= \rho_1^2 C_1(\mathcal{Z}_2, \mathcal{Z}_2) + C_2(\mathcal{Z}_2, \mathcal{Z}_2) \end{aligned}$$

The covariance between models is

$$\begin{aligned} \mathbb{C}\text{ov}\left[f_1(\mathcal{Z}_1), f_2(\mathcal{Z}_2)\right] &= \mathbb{C}\text{ov}\left[\delta_1(\mathcal{Z}_1), \rho_1 \delta_1(\mathcal{Z}_2) + \delta_2(\mathcal{Z}_2)\right] \\ &= \mathbb{C}\text{ov}\left[\delta_1(\mathcal{Z}_1), \rho_1 \delta_1(\mathcal{Z}_2)\right] = \rho_1 C_1(\mathcal{Z}_1, \mathcal{Z}_2) \end{aligned}$$

$\rho \in \mathbb{R}$

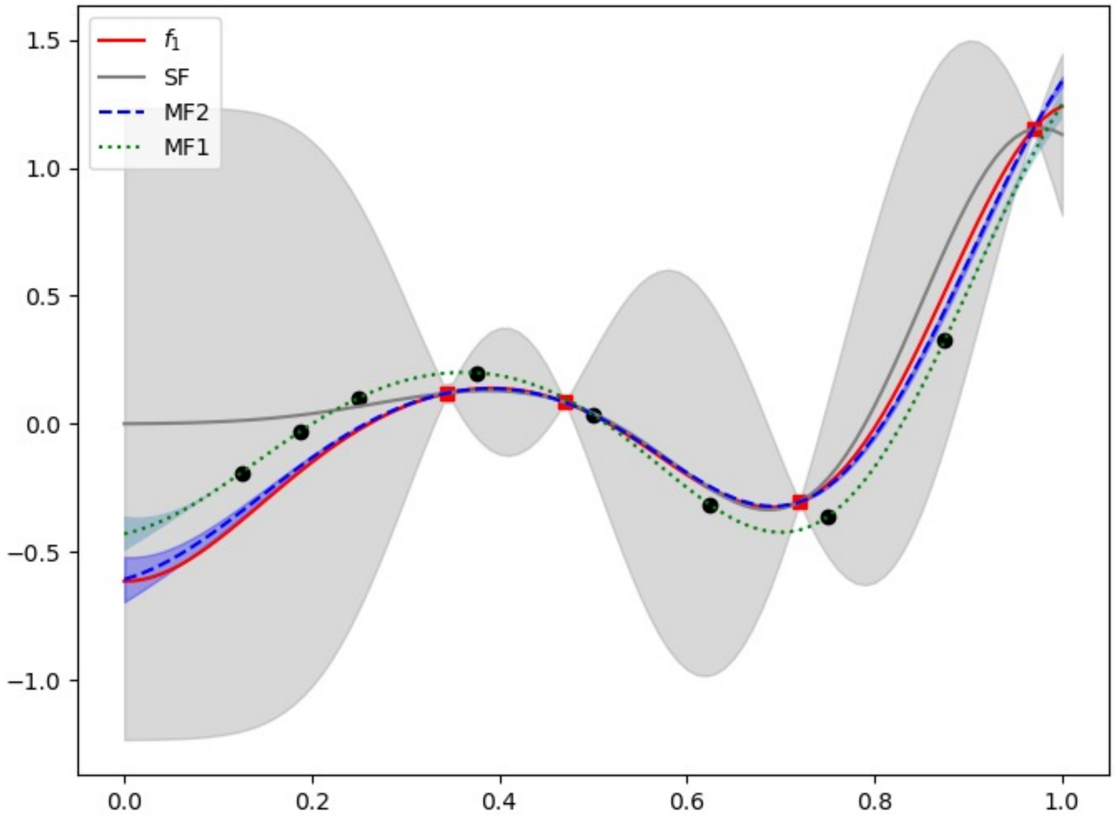
$\rho(z) = \rho_0 + \rho_1 z$

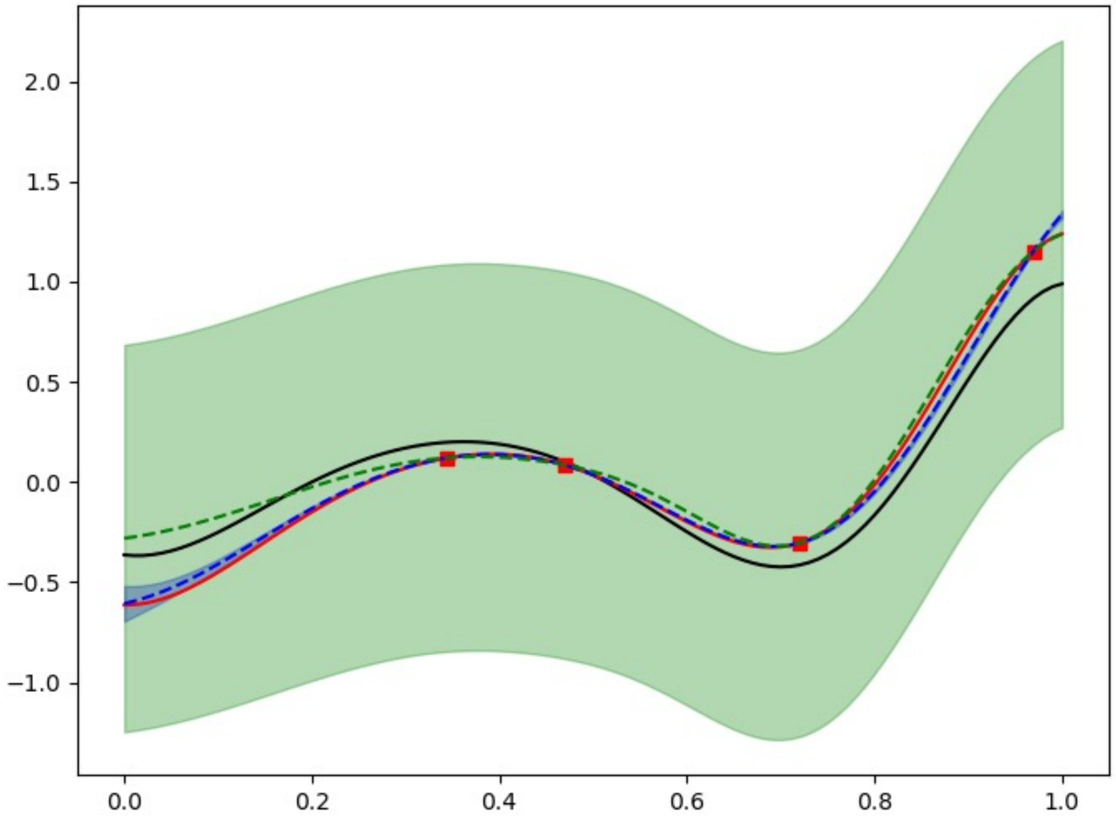


Multilevel Gaussian Processes

The multilevel GP is a better approximation than the single fidelity GP using only the HF data

Alternative methods build GP sequentially similar to multilevel collocation. However there is no way to estimate error consistently and the resulting GP is often less accurate



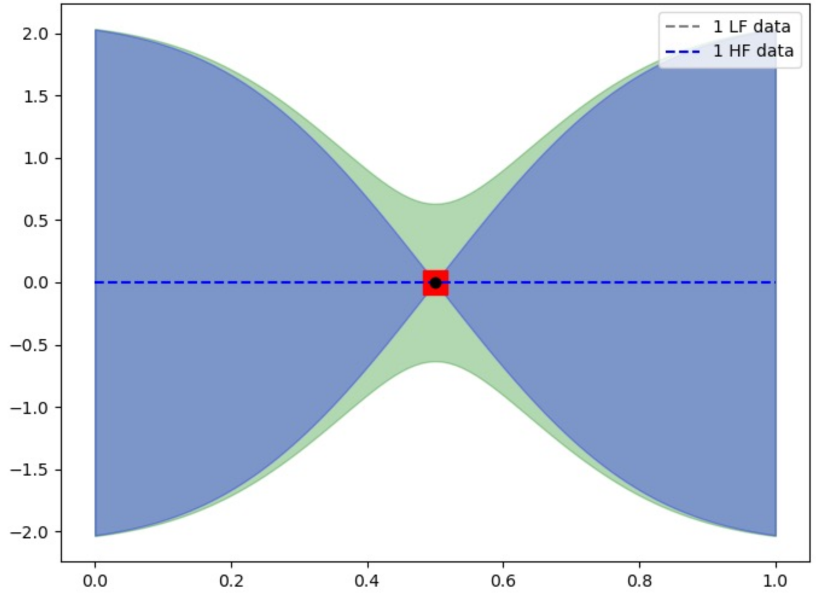


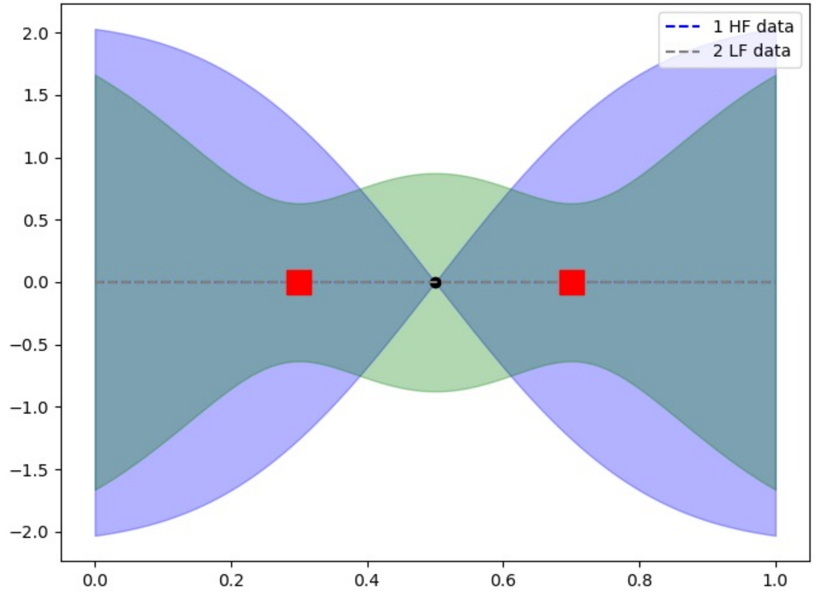
ML GP Experimental Design

Similar to single-fidelity GPs, not all training data reduce variance equally.

But ML GPs have the additional complication that function data evaluated using different models at the same sample z reduce uncertainty differently.

The relative cost of evaluating each model must also be accounted for. Two low-fidelity evaluations may be more cost effective at reducing variance in HF prediction

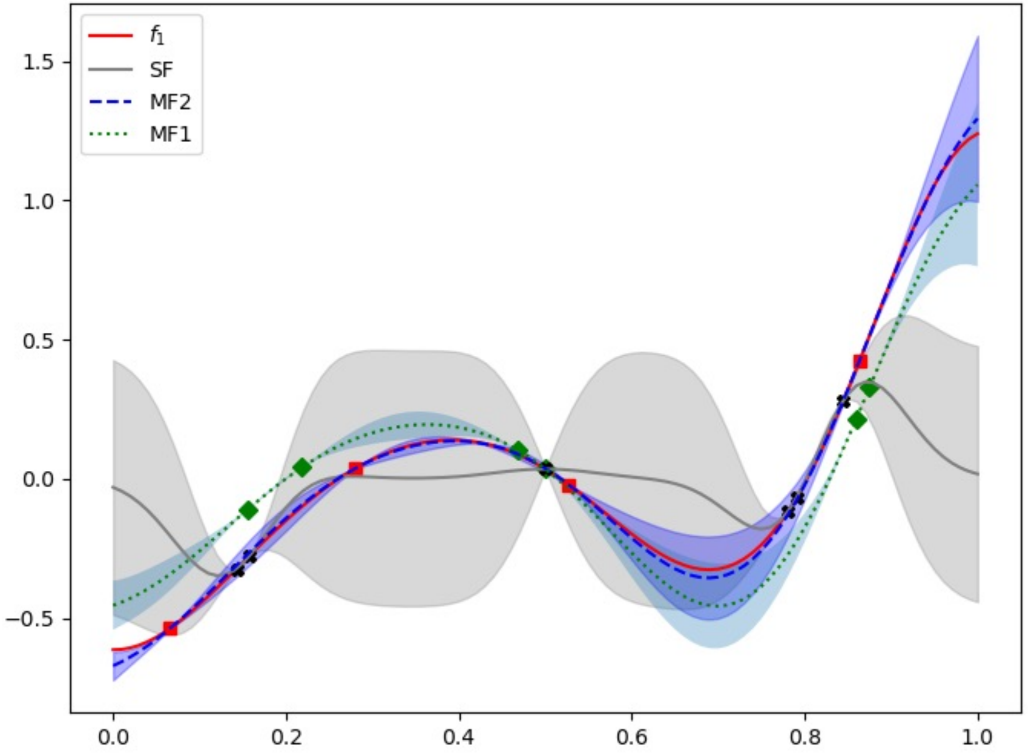




L. Le Gratiet and J. Garnier Recursive co-kriging model for design of computer experiments with multiple levels of fidelity. International Journal for Uncertainty Quantification, 4(5), 365–386, 2014.

ML GP Experimental Design

Assume cost of evaluating each model is $W_1 = 1, W_2 = 3$



The ML-GP is much more accurate for the same amount of work

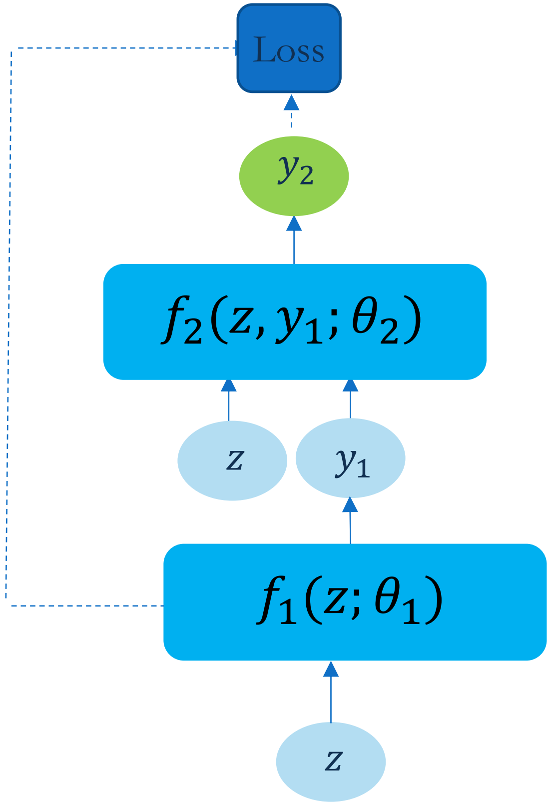
NONLINEAR MULTI-FIDELITY SURROGATES

Multilevel GPs assume a linear relationship between models

$$f_m(z) = \rho_{m-1} f_{m-1}(z) + \delta_m(z)$$

However, a nonlinear model may more efficiently capture the relationship.

$$f_2(z) = g(f_1(z))$$



Deep Gaussian processes where g and f_1 are both Gaussian process have been proposed

Using neural networks for g and f_1 has also been proposed

P. Perdikaris, et al. Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling. Proceedings of the Royal Society of London A. 2017

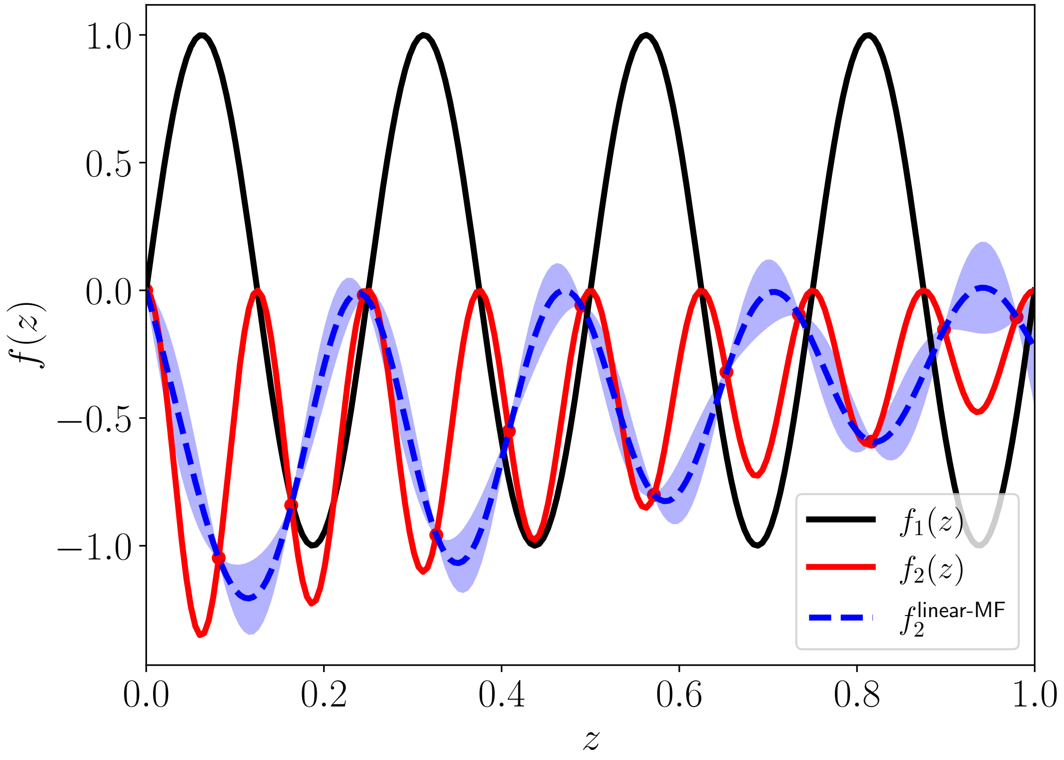
K. Cutajar et al. Deep Gaussian Processes for Multi-fidelity Modeling. 2019,

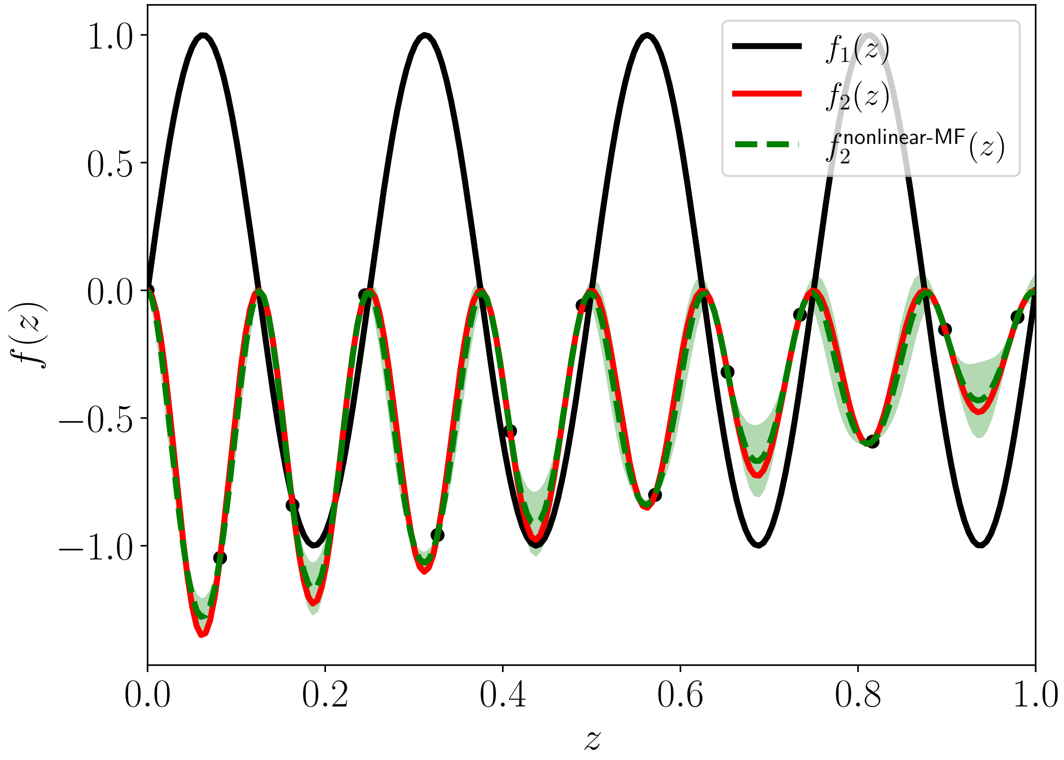
Motamed, Mohammad. "A multi-fidelity neural network surrogate sampling method for uncertainty quantification." International Journal for Uncertainty Quantification 10.4 (2020).

X.Meng, G. Karniadakis, A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse PDE problems, Journal of Computational Physics, (2020)

MF Deep Gaussian Processes

$$f_1(z) = \sin(8\pi z)$$
$$f_2(z) = \left(x - \sqrt{2}\right) f_1(z)^2$$

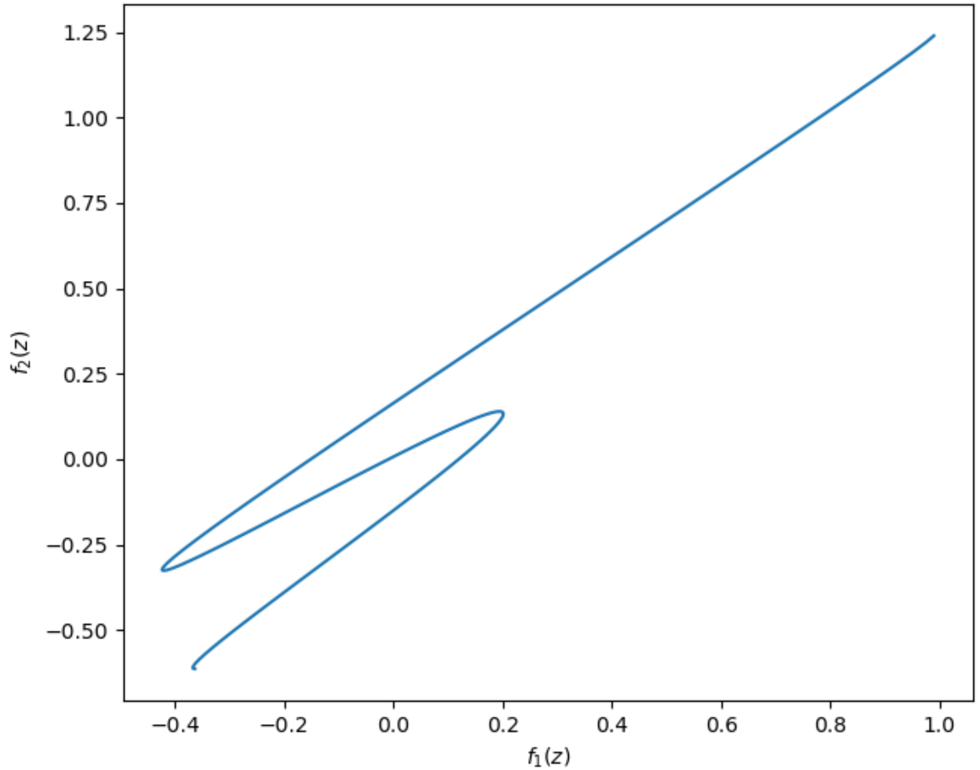


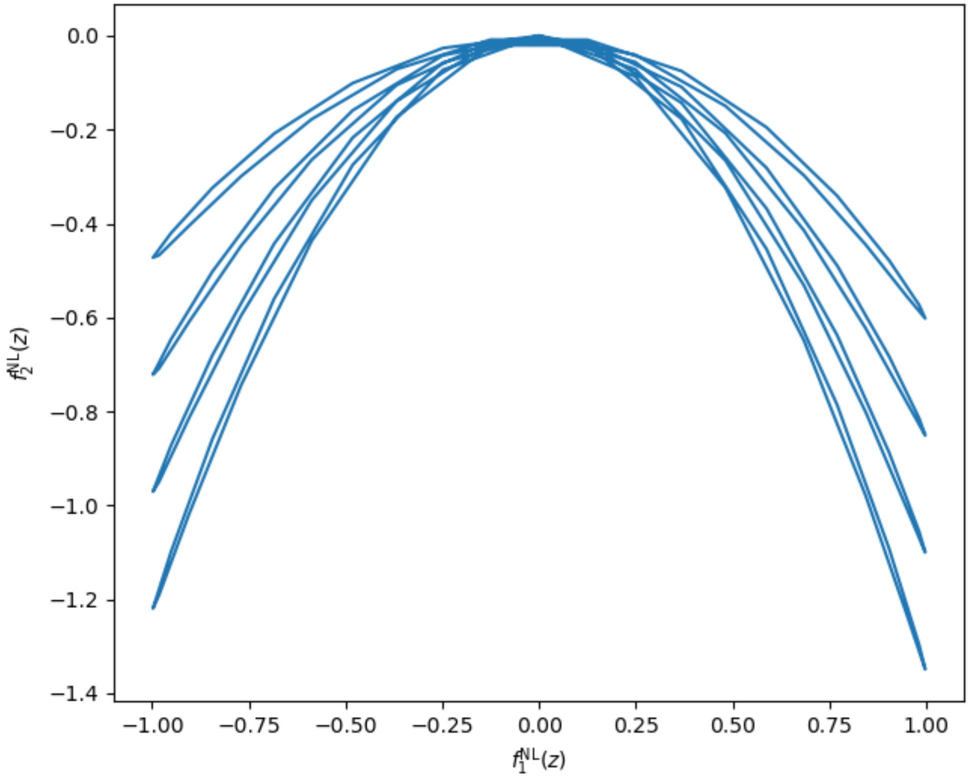


Plots generated using Emukit https://github.com/EmuKit/emukit/

MF DEEP GAUSSIAN PROCESSES

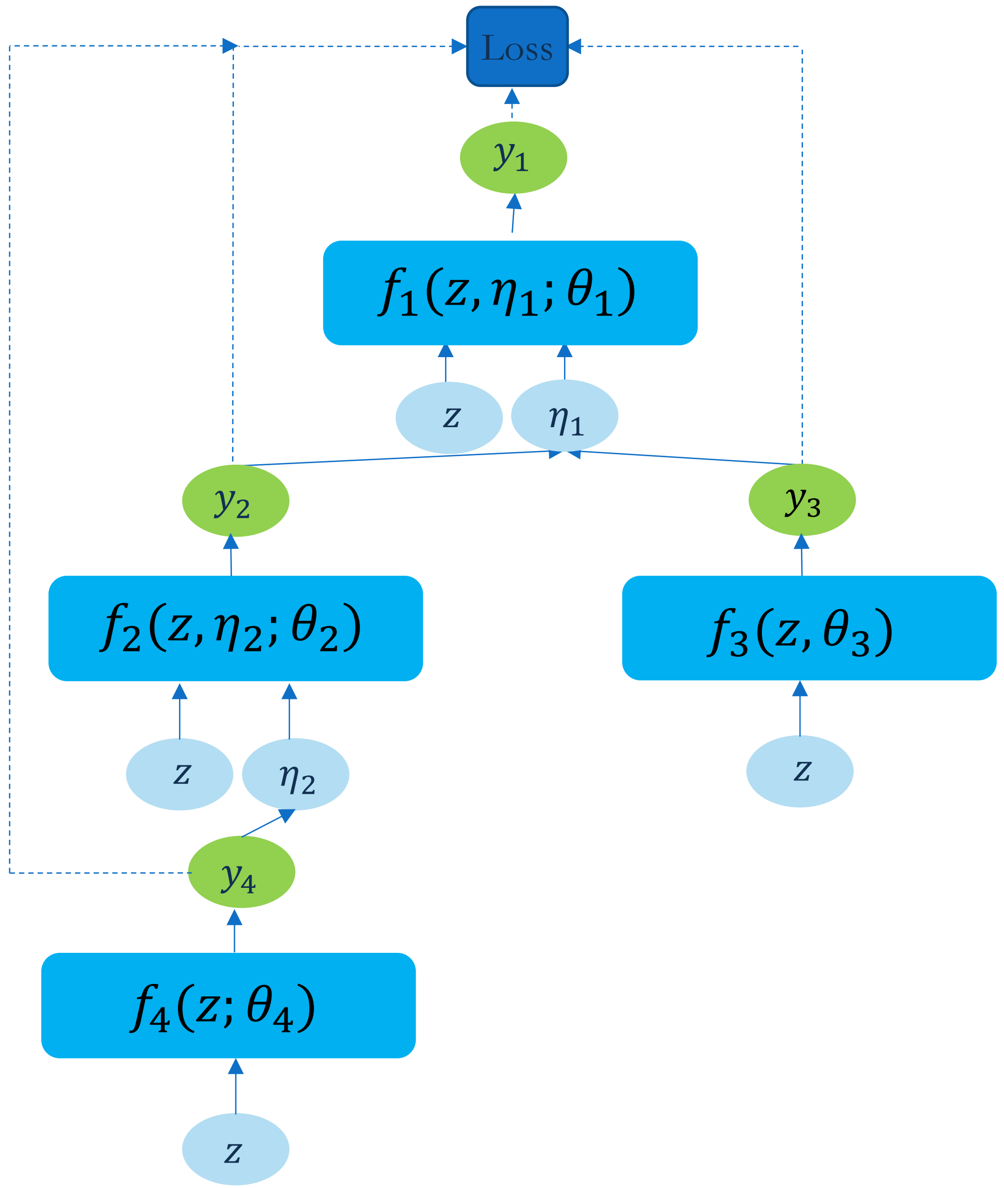
Linear MF GPs tend to perform worse than the non-linear MF GPs when the correlation between models is highly complicated





Non-hierarchical Surrogates

Often models do not admit a 1D hierarchy. In this case we can build multi-fidelity surrogates for models with relationships represented by directed acyclic graphs

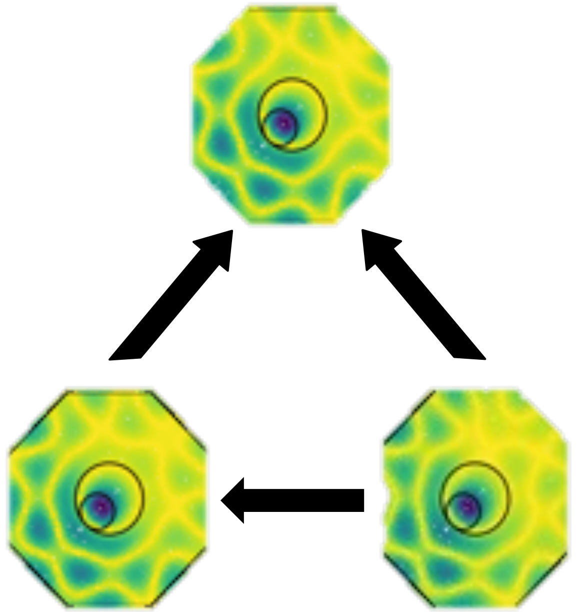


Gorodetsky et al. MFNets: Multi-fidelity data-driven networks for bayesian learning and prediction, International Journal for Uncertainty Quantification, 2020.

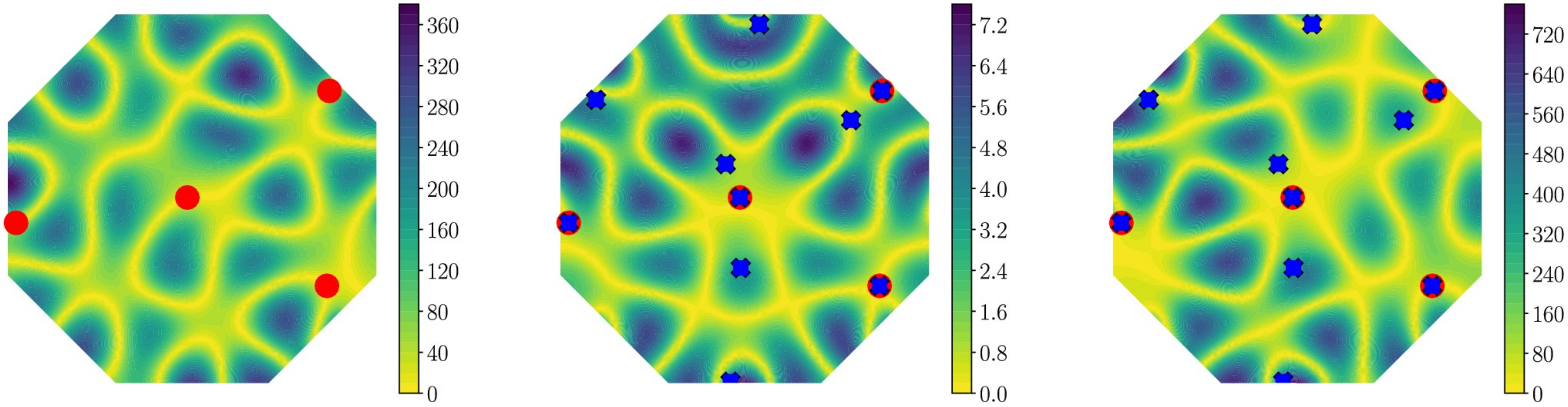
A. Gorodetsky et al. MFNets: Learning network representations for multifidelity surrogate modeling, 2020.

Direct Field Acoustic testing

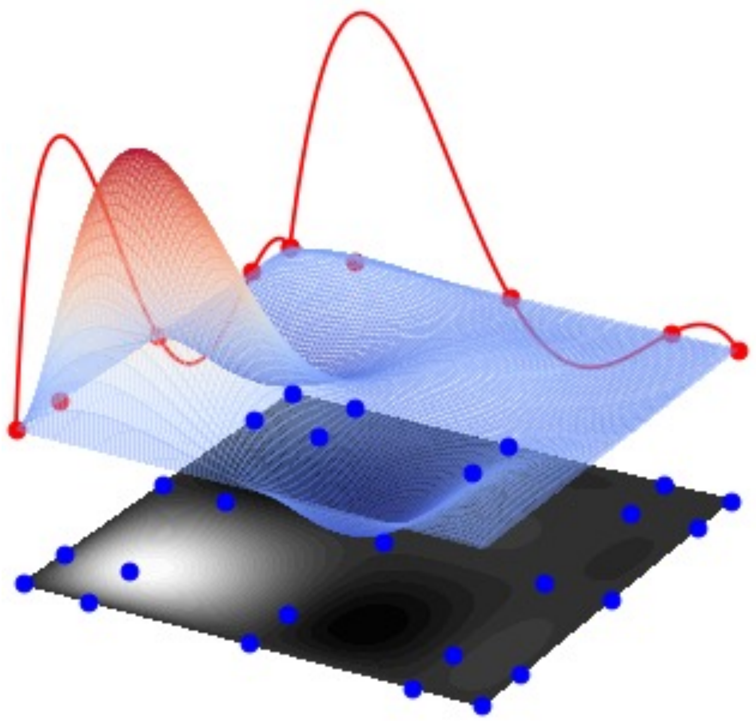
We can fuse multiple experiments that characterize performance of engineered structures under extreme vibration environments simulated using the Helmholtz equation



$$\Delta\phi(x) + \kappa^2\phi(x) = 0\,, \qquad x \in D$$
$$\frac{\partial\phi(x)}{\partial n} = \rho_0 \omega z_i, \qquad x \in \partial D$$



PyApprox



Repository
https://github.com/sandialabs/pyapprox/actions
Documentation
https://sandialabs.github.io/pyapprox/index.html

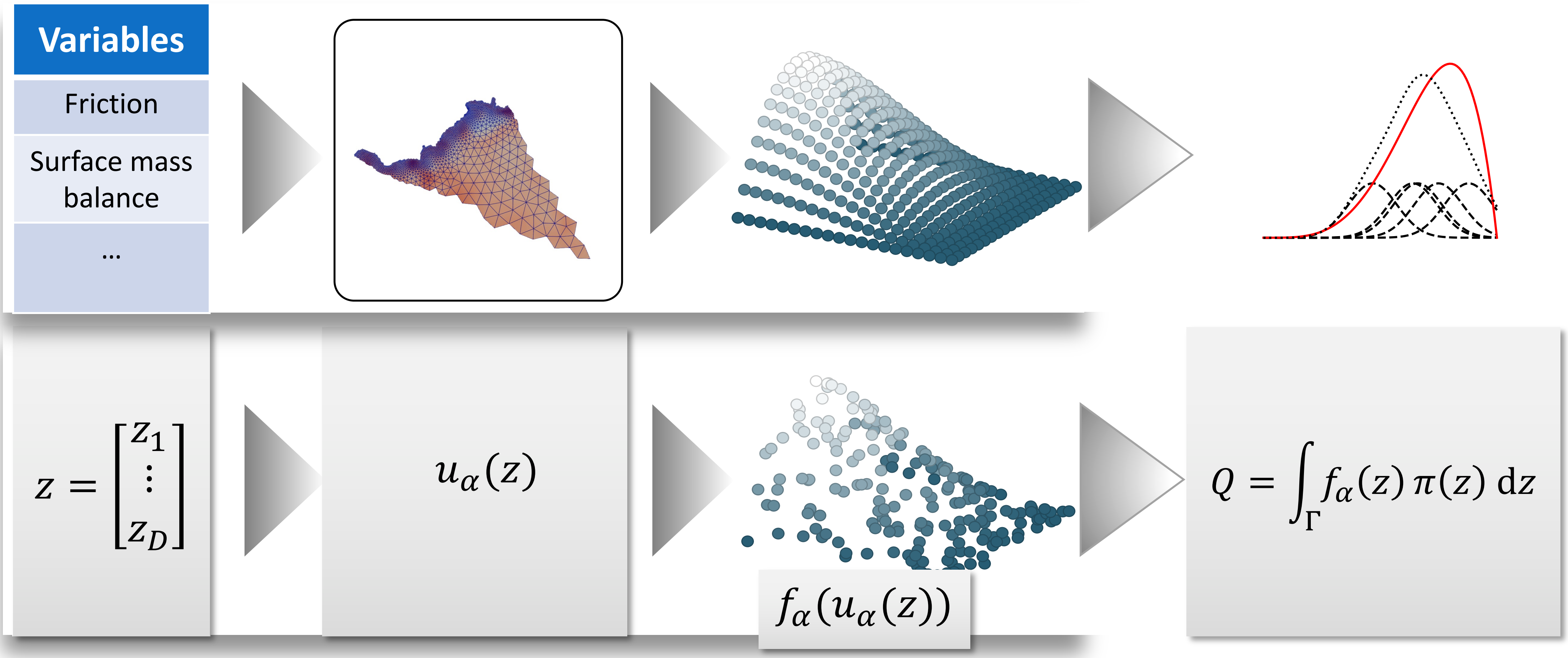
License MIT

Installation Installation of PyApprox and its dependencies managed by Pip or Pip+Anaconda

Target platforms PyApprox is currently built and tested on multiple platforms

Automated Testing: Over 550 tests run on each commit to master

Forward propagation of uncertainty



Must compute statistics from limited number of samples (simulations)
So build surrogate that can be used in place of the expensive model