Report No.
UCB/SEMM-2024/02

Structural Engineering
Mechanics and Materials

Physics-based linear regression for high-dimensional forward uncertainty quantification

By

Ziqi Wang

May 2024

Department of Civil and Environmental Engineering
University of California, Berkeley

Physics-based linear regression for high-dimensional forward uncertainty quantification

Ziqi Wang[a,*]

[a] *Department of Civil and Environmental Engineering, University of California, Berkeley, United States*

Abstract

We introduce linear regression using physics-based basis functions optimized through the geometry of an inner product space. This method addresses the challenge of surrogate modeling with high-dimensional input, as the physics-based basis functions encode problem-specific information. We demonstrate the method using a proof-of-concept nonlinear random vibration example.

Keywords: high-dimensional regression, physics-based surrogate modeling, uncertainty quantification

1. Problem statement and introduction

Consider an end-to-end computational model $\mathcal{M} : \boldsymbol{x} \in \mathbb{R}^n \mapsto y \in \mathbb{R}$ that maps an n-dimensional input $\boldsymbol{x}$ into a 1-dimensional output y. The input $\boldsymbol{x}$ is an outcome of a high-dimensional random vector $\boldsymbol{X}$, defined in the probability space $(\mathbb{R}^n, \mathcal{B}_n, \mathbb{P}_{\boldsymbol{X}})$, where $\mathcal{B}_n$ is the Borel σ-algebra on $\mathbb{R}^n$, and $\mathbb{P}_{\boldsymbol{X}}$ is the probability measure of $\boldsymbol{X}$. The random output Y is associated with a probability space $(\mathbb{R}, \mathcal{B}, \mathbb{P}_Y)$, where $\mathbb{P}_Y$ is the push-forward measure of $\mathbb{P}_{\boldsymbol{X}}$ induced by $\mathcal{M}$. We seek a surrogate model $\hat{\mathcal{M}} : \boldsymbol{x} \in \mathbb{R}^n \mapsto \hat{y} \in \mathbb{R}$ to approximate the statistical properties of Y. This task is challenging due to the high dimensionality of $\boldsymbol{x}$. Specifically, conventional data-fitting surrogate models such as polynomial chaos expansion [1–4] and Gaussian process (Kriging) [5–7] become increasingly ineffective as the number of input variables increases. Injecting domain/problem-specific prior knowledge into the surrogate modeling process has proven to be a promising approach to mitigate the curse of dimensionality, as reflected in the advancements in scientific machine learning [8–10] and multi-fidelity uncertainty quantification [11–13]. This short communication adapts and reformulates the recent works [12, 13] on physics-based surrogate modeling into a simple,

*Corresponding author

Email address: ziqiwang@berkeley.edu (Ziqi Wang)

unified framework of linear regression. In this linear regression, the basis functions are simplified physics-based models with tuning parameters. An inner product space is introduced to facilitate the training of these physics-based basis functions. A proof-of-concept example is presented to demonstrate the proposed approach.

2. Linear regression using physics-based basis functions

We represent the surrogate model $\hat{\mathcal{M}}$ by the linear regression:

$$\hat{\mathcal{M}}(\boldsymbol{x}) = \boldsymbol{s}(\boldsymbol{x})\boldsymbol{w}\,, \tag{1}$$

where $\boldsymbol{s}(\boldsymbol{x}) = [s_1(\boldsymbol{x}), s_2(\boldsymbol{x}), \ldots, s_m(\boldsymbol{x}), 1]$ is a row vector of basis functions and $\boldsymbol{w} = [w_1, w_2, \ldots, w_{m+1}]^\intercal$ is a column vector of weights. We define $s_i(\boldsymbol{x})$, $i = 1, 2, \ldots, m$, as physics-based models with tuning parameters $\boldsymbol{\theta}_i$. Note that the basis function vector $\boldsymbol{s}(\boldsymbol{x})$ contains a constant basis as its last component.

We define the inner product between functions of $\boldsymbol{x}$ as:

$$\langle f, g\rangle := \mathbb{E}_{\boldsymbol{X}}\left[\kappa\left(f(\boldsymbol{X}), g(\boldsymbol{X})\right)\right]\,, \tag{2}$$

where $f\,, g : \mathbb{R}^n \mapsto \mathbb{R}$ are functions of $\boldsymbol{x}$, and $\kappa : \mathbb{R}\times\mathbb{R} \mapsto \mathbb{R}$ is a symmetric and positive definite kernel function that induces an inner product in the function space of f or in an implicit feature space of $\phi(f)$, known as the kernel trick [14]. If the kernel is linear, i.e., $\kappa(af + bf', g) = a\kappa(f, g) + b\kappa(f', g)$ for scalars a and b, the inner product is defined in the original function space of f; otherwise, the inner product is formulated in an implicit feature space induced by the kernel. To ensure a finite inner product, we require $\kappa(f, f)$ be integrable with respect to the measure $\mathbb{P}_{\boldsymbol{X}}$.

Given a training set $\mathcal{D} = \{(\boldsymbol{x}^{(i)}, \mathcal{M}(\boldsymbol{x}^{(i)}))\}_{i=1}^N$, we optimize $s_1(\boldsymbol{x}; \boldsymbol{\theta}_1)$ by solving:

$$\boldsymbol{\theta}_1^\star = \arg\max_{\boldsymbol{\theta}} \frac{\langle \mathcal{M}, s_1(\boldsymbol{\theta})\rangle}{\sqrt{\langle \mathcal{M}, \mathcal{M}\rangle\,\langle s_1(\boldsymbol{\theta}), s_1(\boldsymbol{\theta})\rangle}}\,, \tag{3}$$

where the inner product terms can be evaluated by the sample estimates using $\mathcal{D}$. This optimization aims to align s_1 with the direction of $\mathcal{M}$, in the inner product space defined by Eq. (2).

Given $s_1(\boldsymbol{x}; \boldsymbol{\theta}_1^\star)$, we optimize $s_2(\boldsymbol{x}; \boldsymbol{\theta}_2)$ by solving:

$$\boldsymbol{\theta}_2^\star = \arg\max_{\boldsymbol{\theta}} \frac{\langle \mathcal{M} - \mathcal{P}_{s_1}(\mathcal{M}), s_2(\boldsymbol{\theta})\rangle}{\sqrt{\langle \mathcal{M} - \mathcal{P}_{s_1}(\mathcal{M}), \mathcal{M} - \mathcal{P}_{s_1}(\mathcal{M})\rangle\,\langle s_2(\boldsymbol{\theta}), s_2(\boldsymbol{\theta})\rangle}}\,, \tag{4}$$

where $\mathcal{P}_{s_1}(\mathcal{M})$ denotes the projection of $\mathcal{M}$ onto s_1, defined as:

$$\mathcal{P}_{s_1}(\mathcal{M}) := \frac{\langle \mathcal{M}, s_1\rangle}{\langle s_1, s_1\rangle} s_1\,. \tag{5}$$

Eq. (4) aims to align s_2 with the direction of the orthogonal residual, $\mathcal{M}-\mathcal{P}_{s_1}(\mathcal{M})$, between $\mathcal{M}$ and s_1.

Subsequently, given $s_1(\boldsymbol{x};\boldsymbol{\theta}_1^\star), s_2(\boldsymbol{x};\boldsymbol{\theta}_2^\star), ..., s_{j-1}(\boldsymbol{x};\boldsymbol{\theta}_{j-1}^\star)$, the j-th basis function is obtained from:

$$\boldsymbol{\theta}_j^\star = \arg\max_{\boldsymbol{\theta}} \frac{\left\langle \mathcal{M} - \sum_{k=1}^{j-1}\mathcal{P}_{s_k}(\mathcal{M}), s_j(\boldsymbol{\theta}) \right\rangle}{\sqrt{\langle \mathcal{M} - \sum_{k=1}^{j-1}\mathcal{P}_{s_k}(\mathcal{M}), \mathcal{M} - \sum_{k=1}^{j-1}\mathcal{P}_{s_k}(\mathcal{M})\rangle \, \langle s_j(\boldsymbol{\theta}), s_j(\boldsymbol{\theta})\rangle}}\,, \tag{6}$$

where $\mathcal{P}_{s_k}(\mathcal{M})$ denotes the projection of $\mathcal{M}$ onto s_k, expressed by replacing s_1 in Eq. (5) with s_k.

Provided with the optimized physics-based basis functions and appended by a constant basis, their weights can be computed using the conventional linear regression solution:

$$\boldsymbol{w} = (\mathcal{S}^\intercal \mathcal{S})^{-1}\mathcal{S}^\intercal \boldsymbol{\mathcal{Y}}\,, \tag{7}$$

where $\mathcal{S}$ is an $N \times (m+1)$ matrix with entries $\mathcal{S}_{ij} = s_j(\boldsymbol{x}^{(i)})$, in which $s_{m+1}(\boldsymbol{x}) \equiv 1$, and $\boldsymbol{\mathcal{Y}}$ is an $N \times 1$ vector with components $\mathcal{Y}_i = \mathcal{M}(\boldsymbol{x}^{(i)})$, in which $\boldsymbol{x}^{(i)}$ and $\mathcal{M}(\boldsymbol{x}^{(i)})$ are from the training set $\mathcal{D}$. A more general formulation for the weights is $\boldsymbol{w} = \arg\min\langle \mathcal{M}(\boldsymbol{x}) - \boldsymbol{s}(\boldsymbol{x})\boldsymbol{w}, \mathcal{M}(\boldsymbol{x}) - \boldsymbol{s}(\boldsymbol{x})\boldsymbol{w}\rangle$, which is identical to the linear regression solution if we use a simple linear kernel $\kappa(f,g) = fg$.

The number of physics-based basis functions can be adaptively determined: starting with s_1 and gradually increasing the basis until the training error no longer shows significant reduction.

3. A proof-of-concept nonlinear random vibration example

Consider a Duffing oscillator subjected to Gaussian white noise excitation:

$$\ddot{z}(t) + 2\zeta\omega_n\dot{z}(t) + \omega_n^2 z(t) + \beta z^3(t) = a(t)\,, \tag{8}$$

where $\zeta = 0.05$, $\omega_n = 10\,\mathrm{rad/s}$, and $\beta = 2000\,\mathrm{m}^{-2}\mathrm{s}^{-2}$. The Gaussian white noise $a(t)$ has a unit spectral intensity. For numerical simulations, we set a cutoff angular frequency of $30\pi\,\mathrm{rad/s}$ and represent $a(t)$ by 200 independent standard Gaussian random variables $\boldsymbol{X}$ weighted by a Fourier series [15]. The input of the end-to-end model $\mathcal{M}$ is outcomes of the 200-dimensional random vector $\boldsymbol{X}$, while the output is the peak absolute displacement $y = \sup_{t\in[0,10]}|z(t)|$ for a duration of 10 seconds.

We design the initial physics-based basis function s_1 by generalizing the first-order perturbation of Eq. (8):

$$s_1(\boldsymbol{\theta}) = \sup_{t\in[0,10]} \left| h(t;\theta_1,\theta_2) * a(t) - \theta_3\, h(t;\theta_1,\theta_2) * \left(h(t;\theta_1,\theta_2) * a(t)\right)^3 \right|\,, \tag{9}$$

where "$*$" denotes convolution, $h(t;\theta_1,\theta_2)$ is the impulse response function for a single-degree-of-freedom linear system parameterized by the natural frequency $\theta_1 \geq 0$ and damping ratio $\theta_2 \geq 0$, and $\theta_3 \geq 0$ is another tuning parameter controlling the contribution of the first-order perturbation. The zeroth order perturbation term, $h(t) * a(t)$, describes the response of a tunable linear system subjected to $a(t)$. The first-order perturbation term, $h(t) * (h(t) * a(t))^3$, describes the response of the same linear system subjected to the cube of the zeroth order perturbation term.

The basis functions s_i, $i > 1$, are modeled by linearization—Eq. (9) with $\theta_3 = 0$. If m physics-based basis functions are used, the total number of tuning parameters is $2(m-1)+3$. Therefore, the surrogate representation is parsimonious for this high-dimensional problem with a white noise input. Two convolutions are required to evaluate s_1, while one convolution is needed for each s_i, $i = 2, 3, \dots, m$. Thus, $m + 1$ convolutions are required to evaluate the basis function vector once, which takes negligible time for a relatively small m.

We optimize the physics-based basis functions using the formulas in Section 2 and a training set of only 30 samples from random realizations of the white noise and their corresponding peak absolute responses. To define the inner product, a simple linear kernel $\kappa(f, g) = fg$ is used. An alternative is $\kappa(f, g) = (f - \mathbb{E}_{\boldsymbol{X}}[f(\boldsymbol{X})])(g - \mathbb{E}_{\boldsymbol{X}}[g(\boldsymbol{X})])$, which is also linear. We did not find significant evidence to favor one option over the other, so either can be used effectively. Note that nonlinear kernels can also be employed, but there is no theoretical reason to favor them in this particular example. The number of physics-based basis functions is adaptively determined, such that by increasing the basis, the reduction in the mean square relative training error is less than 1%. This number typically ranges from 2 to 4 due to variations in the training set. Using a test set of 10^4 random samples, Figure 1 compares the reference Runge–Kutta–Fehlberg solutions with the surrogate model predictions. Figure 2 illustrates the basis function responses.

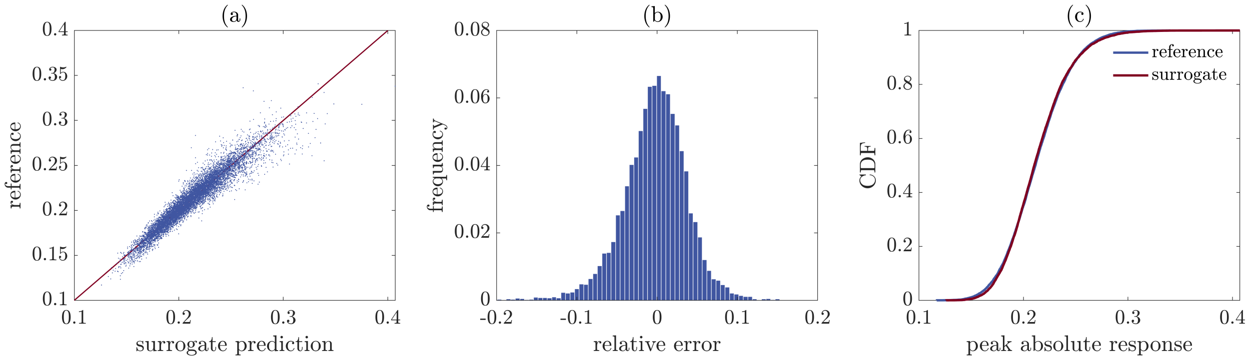

Figure 1: **Performance of the physics-based linear regression: (a) scatter plot of the reference solutions against surrogate model predictions, (b) histogram of the relative error, and (c) prediction on cumulative distribution function**. The reference solutions for the peak absolute responses are obtained from the Runge–Kutta–Fehlberg method. The surrogate model predictions are obtained from Eq. (1) using a training set of only 30 samples. For the simulation results reported in this figure, the mean relative error is -0.3%, the first quartile is -2.5%, and the third quartile is 2.3%. Thus, on average, the surrogate model slightly overestimates the peak absolute response.

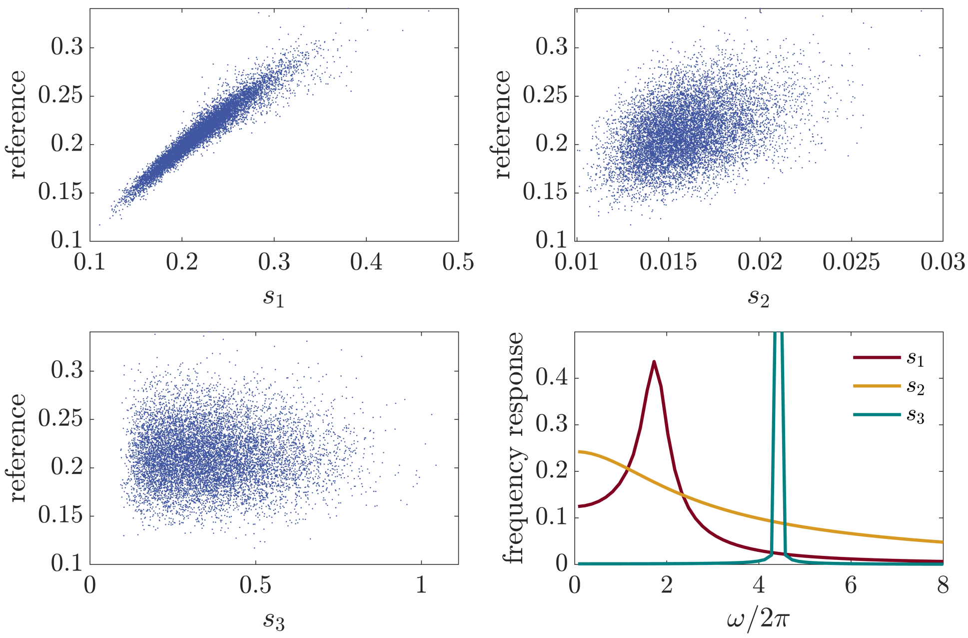

Figure 2: **Basis function responses**. The top-left panel shows the reference peak absolute responses against the outputs of the basis function s_1, evaluated using a test set of 10^4 samples. The top-right and bottom-left panels illustrate the scenarios for s_2 and s_3, respectively. The bottom-right panel displays the frequency response functions (FRFs) associated with the three basis functions. It is seen that s_1 is mostly correlated with $\mathcal{M}$, as expected from the goal of Eq. (3), while s_2 and s_3 are optimized to fill the orthogonal residues. The FRFs clearly illustrate the differences in these basis functions. The FRF for s_1 is obtained from the zeroth-order perturbation term.

4. Conclusion

This short communication introduces linear regression using physics-based basis functions. The main contribution is to standardize the procedure for training physics-based basis functions. A proof-of-concept random vibration example of a Duffing oscillator demonstrates the potential of this approach. Future research directions may include extensions to multivariate output models, adaptations for rare-event probability estimations, and applications in inverse uncertainty quantification, sensitivity, and optimization problems.

References

[1] C. Soize, R. Ghanem, Physical systems with random uncertainties: chaos representations with arbitrary probability measure, SIAM Journal on Scientific Computing 26 (2) (2004) 395–410.

[2] D. Xiu, Numerical methods for stochastic computations: a spectral method approach, Princeton university press, 2010.

[3] E. Torre, S. Marelli, P. Embrechts, B. Sudret, Data-driven polynomial chaos expansion for machine learning regression, Journal of Computational Physics 388 (2019) 601–623.

[4] L. Novák, H. Sharma, M. D. Shields, Physics-informed polynomial chaos expansions, Journal of Computational Physics 506 (2024) 112926.

[5] P. K. Kitanidis, Introduction to geostatistics: applications in hydrogeology, Cambridge university press, 1997.

[6] K. P. Murphy, Machine learning: a probabilistic perspective, MIT press, 2012.

[7] X. Yang, D. Barajas-Solano, G. Tartakovsky, A. M. Tartakovsky, Physics-informed cokriging: A gaussian-process-regression-based multifidelity method for data-model convergence, Journal of Computational Physics 395 (2019) 410–431.

[8] M. Raissi, P. Perdikaris, G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, Journal of Computational Physics 378 (2019) 686–707.

[9] Y. Zhu, N. Zabaras, P.-S. Koutsourelakis, P. Perdikaris, Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data, Journal of Computational Physics 394 (2019) 56–81.

[10] A. F. Psaros, X. Meng, Z. Zou, L. Guo, G. E. Karniadakis, Uncertainty quantification in scientific machine learning: Methods, metrics, and comparisons, Journal of Computational Physics 477 (2023) 111902.

[11] B. Peherstorfer, K. Willcox, M. Gunzburger, Survey of multifidelity methods in uncertainty propagation, inference, and optimization, Siam Review 60 (3) (2018) 550–591.

[12] Z. Wang, Optimized equivalent linearization for random vibration, Structural Safety 106 (2024) 102402.

[13] J. Xian, Z. Wang, A physics and data co-driven surrogate modeling method for high-dimensional rare event simulation, Journal of Computational Physics (2024) 113069.

[14] T. Hofmann, B. Schölkopf, A. J. Smola, Kernel methods in machine learning, The Annals of Statistics 36 (3) (2008) 1171–1220.

[15] C. Chatfield, The analysis of time series: theory and practice, Springer, 2013.