
Bayesian Polynomial Chaos

Pranay Seshadri

Department of Mathematics
Imperial College London
London, SW7 2AZ
p.seshadri@imperial.ac.uk

Andrew Duncan

Department of Mathematics
Imperial College London
London, SW7 2AZ
a.duncan@imperial.ac.uk

Ashley Scillitoe

Data-Centric Engineering
The Alan Turing Institute
London, NW1 2DB
ascillitoe@turing.ac.uk

Abstract

In this brief paper we introduce Bayesian polynomial chaos, a Gaussian process analogue to polynomial chaos. We argue why this Bayesian re-formulation of polynomial chaos is necessary and then proceed to mathematically define it, followed by an examination of its utility in computing moments and sensitivities; multi-fidelity modelling, and information fusion.

1 Introduction & motivation

Over the past decade, polynomial chaos [1] has garnered significant industrial uptake within engineering [2, 3, 4, 5]. Tailored as an approach to *aleatory uncertainty quantification*, its simplicity, both expositional and practical, has cemented it as a must-have tool in many computational design, health monitoring, and manufacturing workflows. At the expositional level, polynomial chaos requires uncertain inputs to computational models be characterized by probability distributions. These uncertainties are then *propagated* through the aforementioned models by non-intrusively (although recipes for intrusive polynomial chaos exist within literature, see [6]) evaluating the model at different input conditions, adhering to a design of experiment set by numerical quadrature rules, such as Gauss-Christoffel, Clenshaw-Curtis, and many more—through tensorial, sparse grid [7, 8], least squares [9] and compressive sensing [10] approaches. The objective of polynomial chaos tools is to quantify the uncertainty in black-box model output quantities of interest (qois). In practical terms, polynomial chaos affords relatively tractable estimates of the moments of output qois and their sensitivities—through Sobol’ indices and related metrics. However, beyond its interpretability, polynomial chaos is purpose-built for working with typically long running engineering models, e.g., computational fluid dynamics and finite elements solutions. Here output qois are by and large spatio-temporal integrals of scalar fields, and input qois are boundary conditions or geometry parameters. Functional representations between the model inputs and outputs are typically smooth and continuous (even though the scalar fields themselves may exhibit discontinuities, e.g., shocks in Eulerian flow), making them ideal candidates for polynomial approximations.

Research into polynomial chaos has been burgeoning with a focus on four distinct disciplines: sampling methods and their underlying distributions [11]; curse of dimensionality alleviation via dimension reduction [12]; tackling correlations in input parameters [13], and estimating higher-order sensitivity information [14, 15]. Given the flurry of academic and industrial activity, it stands to reason that any profound change in overarching methodology must be duly justified. So why a Bayesian variant of polynomial chaos? First, over the past few years there has been a concentrated

effort at quantifying the *epistemic* uncertainties in numerical simulations in both fluids [16] and structures [17]. This has led to a paradigm where a single evaluation of a *black-box* model for a unique input yields a distribution for a chosen output qoi. Second, for most real-world problems, numerical simulations are deployed on cases far more complex (e.g., jet engines and aircrafts) than the canonical ones (e.g., scaled compressor rigs and scaled wind tunnel airfoil tests) used for validating and calibrating their underlying equation systems. Thus, even in the absence of a rigorous framework to quantify epistemic uncertainty, a naïve stratagem for expressing a modeler’s certainty in an output qoi instance, can be conveyed through an assigned probability. This could arise from either changes in convergence, or deviations from well-established validation cases (e.g., stall and transition). Third, and perhaps more philosophically, when experimental data is available, it is typically used to set boundary conditions and to gauge the utility of the numerical simulation tool. It is not used to augment or supplement what happens between the boundaries. This approach, by design, thwarts attempts to fuse sensor data with numerical simulations. We argue that a Bayesian approach can be designed to rigorously combine experimental values and numerical solutions for delivering greater inference; shedding more insight into the observed physical phenomenon.

In this paper, we introduce Bayesian polynomial chaos, a Gaussian process analogue to polynomial chaos. It is effectively a Gaussian process with orthogonal polynomial kernels, which builds upon advances in both machine learning and uncertainty quantification. The remainder of this relatively short paper is structured as follows. First, we present a mathematical treatment of Bayesian polynomial chaos; its computation of moments; utility in multi-fidelity and sensor-fusion efforts. This exposition is sprinkled with a few representative examples. We close this manuscript by elaborating on next steps.

2 Bayesian polynomial chaos

2.1 Preliminaries

We consider the problem of approximating a function $f(\mathbf{x})$ where $\mathbf{x} = (x^{(1)}, \dots, x^{(d)})$ is a point in \mathbb{R}^d . We assume that each of the d parameters in \mathbf{x} are mutually independent random variables with marginal distributions given by $\rho_1(x^{(1)}), \dots, \rho_d(x^{(d)})$ respectively, yielding a joint density given by $\boldsymbol{\rho}(\mathbf{x}) = \rho_1(x^{(1)}) \times \rho_2(x^{(2)}) \times \dots \times \rho_d(x^{(d)})$. In polynomial chaos parlance, $\boldsymbol{\rho}(\mathbf{x})$ characterises the input uncertainty to the model $f(\mathbf{x})$. As we typically do not have access to an analytical form for this model, we approximate it with an orthogonal polynomial expansion

$$f(\mathbf{x}) \approx g(\mathbf{x}) = \sum_{j=1}^N \alpha_j \phi_j(\mathbf{x}), \quad (1)$$

where α_j are the unknown coefficients and $\phi_j(\mathbf{x}) = \prod_{i=1}^d \phi_j^{(i)}(x^{(i)})$ denotes the j -th basis term, which in turn is given as a product of its composite univariate polynomials. We can express the polynomial expansion in Eq. 1 as a parameterised matrix equation of the form

$$g(\mathbf{x}) = \mathbf{V}^T(\mathbf{x}) \boldsymbol{\alpha}, \quad \text{where } \mathbf{V}_{ij} = \phi_j(\mathbf{x}_i) \quad \text{and} \quad \boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_N \end{bmatrix}, \quad (2)$$

with $\mathbf{V} \in \mathbb{R}^{M \times N}$, where M corresponds to the number of model evaluations, i.e., we assume the existence of discrete model evaluations in the form of input-output pairs $\{\mathbf{x}_k, f_k\}_{k=1}^M$.

A few further remarks on the the approximation in Eq. 1 are in order. Here the composition of each multivariate polynomial is set by a *multi-index set* $j = (j^{(1)}, \dots, j^{(d)}) \in \mathbb{N}_0^d$ which determines which univariate polynomial orders participate in the multivariate approximation in Eq. 1. Each univariate polynomial satisfies the property

$$\int_{\mathbb{R}} \phi_a^{(i)}(x^{(i)}) \phi_b^{(i)}(x^{(i)}) \rho_i(x^{(i)}) dx^{(i)} = \delta_{a,b}, \quad (3)$$

where $\delta_{a,b}$ is the Kronecker delta. Eq. 3 is a statement on the orthogonality of the composite polynomials, the existence of which is guaranteed under certain mild technical conditions on ρ_i (see

[18]). As we will see later, orthogonality plays an important role in polynomial chaos as it facilitates easy computation of certain statistical moments. In fact, the fundamental tenet behind polynomial chaos is that selecting orthogonal polynomials (and their corresponding weight functions) enables exponential convergence in the estimation of output moments [1].

2.2 Representation as a Gaussian process

In our Bayesian analogue we think of g , the polynomial approximation, as a Gaussian process

$$g(\mathbf{x}) \sim \mathcal{N}(\mu_g(\mathbf{x}), \Sigma_g(\mathbf{x}, \mathbf{x}')), \quad (4)$$

defined entirely by μ_g and Σ_g , its mean and covariance functions respectively. We assume there exists a prior on the polynomial coefficients that feed into this Bayesian polynomial, given by

$$\mathbf{p}(\boldsymbol{\alpha}^{(0)}) \sim \mathcal{N}(\mu_{\boldsymbol{\alpha}^{(0)}}, \Sigma_{\boldsymbol{\alpha}^{(0)}}), \quad (5)$$

where $\mu_{\boldsymbol{\alpha}^{(0)}} \in \mathbb{R}^N$. Given a likelihood model of the form $\mathbf{p}(\mathbf{V}^T \boldsymbol{\alpha}, \eta \mathbf{I})$, where η is a small numerical nugget, the posterior distribution $\mathbf{p}(\mu_{\boldsymbol{\alpha}}, \Sigma_{\boldsymbol{\alpha}})$ is defined via

$$\Sigma_{\boldsymbol{\alpha}} = \left(\frac{1}{\eta^2} \mathbf{V} \mathbf{V}^T + \Sigma_{\boldsymbol{\alpha}^{(0)}}^{-1} \right)^{-1} \quad \text{and} \quad \mu_{\boldsymbol{\alpha}} = \Sigma_{\boldsymbol{\alpha}} \left(\frac{1}{\eta^2} \mathbf{V} \mathbf{f} + \Sigma_{\boldsymbol{\alpha}^{(0)}}^{-1} \mu_{\boldsymbol{\alpha}^{(0)}} \right), \quad (6)$$

where $\mathbf{f} = (f_1, \dots, f_M)^T$. The mean and covariance functions in Eq. 4 can then be computed to be

$$\mu_g(\mathbf{x}) = \mathbf{V}^T(\mathbf{x}) \mu_{\boldsymbol{\alpha}} \quad \text{and} \quad \Sigma_g(\mathbf{x}, \mathbf{x}') = \mathbf{V}(\mathbf{x})^T \Sigma_{\boldsymbol{\alpha}} \mathbf{V}(\mathbf{x}'). \quad (7)$$

The expressions above are a direct consequence of Bayes' theorem and their detailed derivations can be found in page 124 of [19].

2.3 Computing moments and sensitivities

One of the salient advantages of polynomial chaos is the relatively easy computation of moments and sensitivities. For instance, the mean of the output g with respect to the prescribed input uncertainties is given by the first coefficient of the polynomial approximation; the variance is given by the sum of the squares of the remaining coefficients (see page 210 in [20]). For Bayesian polynomials, as defined in Eq. 4, the moment computations are linear operators acting on the Gaussian process, yielding a Gaussian distribution. The expectation is given by

$$\begin{aligned} \mathbb{E}[g] &= \int_{\mathbb{R}^d} g(\mathbf{x}) \boldsymbol{\rho}(\mathbf{x}) d\mathbf{x} \\ &\sim \mathcal{N} \left(\int_{\mathbb{R}^d} \mathbf{V}^T(\mathbf{x}) \mu_{\boldsymbol{\alpha}} \boldsymbol{\rho}(\mathbf{x}) d\mathbf{x}, \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} \mathbf{V}^T(\mathbf{x}) \Sigma_{\boldsymbol{\alpha}} \mathbf{V}^T(\mathbf{x}') \boldsymbol{\rho}(\mathbf{x}) \boldsymbol{\rho}(\mathbf{x}') d\mathbf{x} d\mathbf{x}' \right), \end{aligned} \quad (8)$$

where standard Gauss-Christoffel quadrature rules of appropriate order can be used to arrive at the exact expressions. In a similar manner, we write the variance as

$$\begin{aligned} \text{Var}[g] &= \int_{\mathcal{X}} g^2(\mathbf{x}) \boldsymbol{\rho}(\mathbf{x}) d\mathbf{x} - (\mathbb{E}[g])^2 \\ &\sim \mathcal{N} \left(\int_{\mathbb{R}^d} (\mathbf{V}^T(\mathbf{x}) \mu_{\boldsymbol{\alpha}})^2 \boldsymbol{\rho}(\mathbf{x}) d\mathbf{x} - \left[\int_{\mathbb{R}^d} \mathbf{V}^T(\mathbf{x}) \mu_{\boldsymbol{\alpha}} \boldsymbol{\rho}(\mathbf{x}) d\mathbf{x} \right]^2, \right. \\ &\quad \left. \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} (\mathbf{V}^T(\mathbf{x}))^2 \Sigma_{\boldsymbol{\alpha}} (\mathbf{V}^T(\mathbf{x}'))^2 \boldsymbol{\rho}(\mathbf{x}) \boldsymbol{\rho}(\mathbf{x}') d\mathbf{x} d\mathbf{x}' \right). \end{aligned} \quad (9)$$

Expressions for conditional variances and the related Sobol' indices can also be derived. For these, one simply constrains terms in \mathbf{V} corresponding to the dimension and order required.

Here too, numerical quadrature comes to the rescue making these seemingly unwieldy integrals available exactly. Figure 1 demonstrates these expressions in action for a bivariate polynomial model deployed on an aerodynamic problem: aleatory uncertainties in the turbulent-to-laminar viscosity $\nu_t/\nu \sim \mathcal{U}[1, 100]$ and turbulence intensity $Ti \sim \mathcal{N}(10, 5)$ are injected into a computational fluid dynamics (CFD) model of a turbine blade, where the impact on performance through its pressure

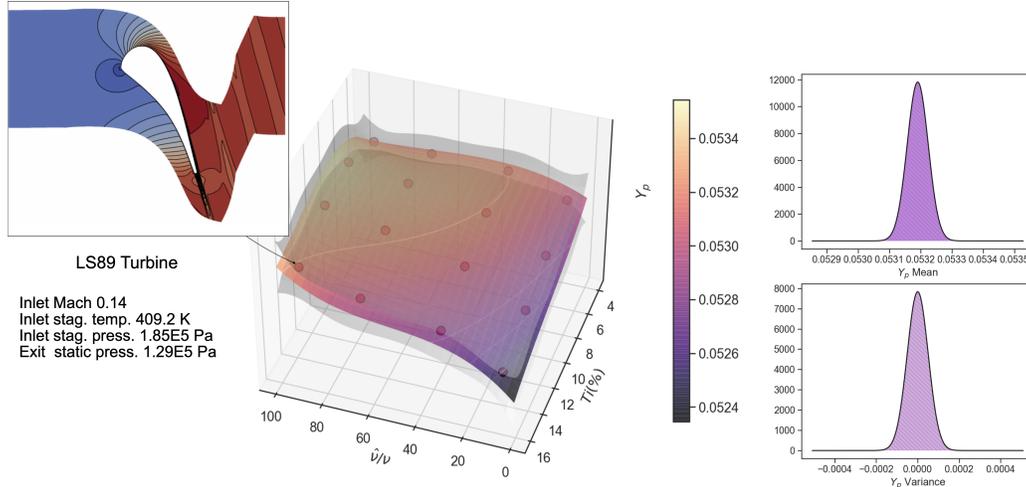


Figure 1: Bayesian polynomial chaos applied on the LS89 turbine with two input aleatory uncertainties: ν_t/ν and T_i .

loss Y_p needs to be determined. We assume that owing to an underpinning epistemic uncertainty in turbulence closures, each evaluation of the model has an uncertainty (standard deviation) of $Y_p \pm 0.01$. The posterior polynomial mean and its standard deviation is shown in Figure 1 along with mean and variance in Y_p , obtained using Eqs. 8 and 9. A third order isotropic bivariate polynomial was used with a tensor grid quadrature rule requiring a total of 16 CFD evaluations. It should be noted that the posterior polynomial is robust to variations in the prior, as the former is entirely set by the model uncertainty ($Y_p \pm 0.01$) and the data, as the data is adequate to determine the unknown polynomial coefficients. However, there are applications where this may not be permissible and greater reliance on the prior may be necessary.

2.4 Multi-fidelity models and design of experiment

An opportunity for precisely this arises in multi-fidelity modeling paradigms. These are characterised by computationally expensive high-fidelity models and relatively inexpensive low-fidelity models. In literature, a common workflow is to approximate an error function between a high-fidelity model output and its low-fidelity counterpart [21]; parsimoniously evaluating the high-fidelity model while extensively sampling the low-fidelity one. Here we propose a different approach, one that arguably carries greater resonance with a Bayesian viewpoint: we use the low-fidelity model’s posterior as the prior ($\mu_{\alpha^{(0)}}, \Sigma_{\alpha^{(0)}}$) for the high-fidelity model. Then, we impute the posterior distribution for the high-fidelity model by sequentially evaluating it at an appropriately selected point x^* . This simple approach is more resistant to errors that are artefacts of numerical noise, as the model’s underlying structure is preserved and propagated a level up. If the data contradicts the prior, then this structure will also be revealed, albeit at the cost of more high-fidelity evaluations. Additionally, this approach easily scales to an arbitrary number of levels, with the prospect of introducing additional constraints into the prior (see forthcoming section). A conceptual illustration is shown in Figure 2 for an underlying cubic model; the low-fidelity model has five evaluations, whilst the high-fidelity model only has one. However, despite this, it still retains the same structure as the low-fidelity owing to its imputed prior.

For cases where the varying models have different inputs—typically the higher the fidelity, the greater d —subspace polynomials of the form $\mathbf{V}(\mathbf{C}^T \mathbf{x})$ can be constructed, where $\mathbf{C}^{d \times n}$ with $n < d$ can be tailored to ensure that all polynomial models adhere to the same input space as the highest-fidelity one. In cases where such spaces are simply infeasible to establish, a coregional approach [22] can also be adopted at the cost of having a more complex covariance structure.

To identify new high-fidelity points, we design an optimisation strategy that maximises the marginal likelihood, $p(\mathbf{f}|\mathbf{V}, \mu_{\alpha^{(0)}}, \Sigma_{\alpha^{(0)}}) = \int p(\mathbf{f}|\mathbf{V}, \mu_{\alpha}, \eta^2) p(\mu_{\alpha}|\mu_{\alpha^{(0)}}, \Sigma_{\alpha^{(0)}}) d\mu_{\alpha}$. In standard Gaus-

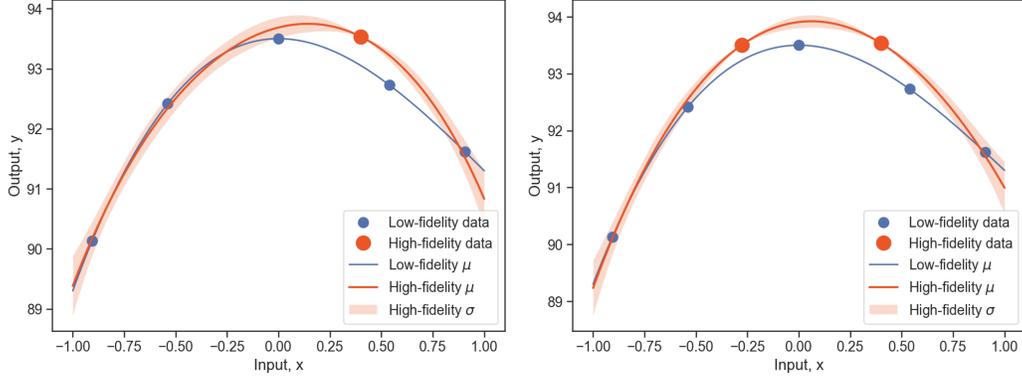


Figure 2: Illustration of a bi-fidelity model where the posterior of the low-fidelity model is used to inform the prior of the high-fidelity model (left). An application of the design of experiment is shown on the right.

sian process fashion, we express this objective as

$$\begin{aligned} & \underset{\mathbf{x}^*}{\text{minimise}} \mathcal{R}(\mathbf{x}^*) \\ & \text{subject to } \mathcal{R}(\mathbf{x}^*) = \frac{M}{2} \log(2\pi) + \frac{M}{2} \log |\Sigma_m| + \frac{1}{2} (\mathbf{f} - \mu_m)^T \Sigma_m^{-1} (\mathbf{f} - \mu_m), \end{aligned} \quad (10)$$

where the mean and covariance of the marginal likelihood are given by $\mu_m = \mathbf{V}^T \mu_{\alpha^{(0)}}$ and $\Sigma_m = \eta^2 \mathbf{I} + \mathbf{V}^T \Sigma_{\alpha^{(0)}} \mathbf{V}$, where \mathbf{V} is augmented with the extra row $\mathbf{V}_i(\mathbf{x}^*)$. Gradients for Eq. 10 can be easily computed and returned with the objective for each new function call in a standard gradient optimisation loop. We share a representative result in Figure 2 using `scipy`'s [23] SLSQP algorithm.

2.5 Fusion of linear-operators

In some engineering applications, it is often easier to estimate integral-, differential-, or more generally linear-operators $\mathcal{L}\{\cdot\}$ of related qois than the required qoi itself. In line with earlier remarks, this data need not arise from simulations but can even be from empirical observations. If probabilistic descriptions of such data is known, then they can be used to constrain the space of posterior polynomial distributions. We formalise this idea by considering the joint distribution

$$\begin{pmatrix} g \\ \mathcal{L}\{g\} \end{pmatrix} = \mathcal{N} \left(\begin{bmatrix} \mu_g \\ \mathcal{L}\{\mu_g\} \end{bmatrix}, \begin{bmatrix} \Sigma_g(\mathbf{x}, \mathbf{x}') & \mathcal{L}\{\Sigma_g(\mathbf{x}, \mathbf{x}')\} \\ \mathcal{L}\{\Sigma_g(\mathbf{x}, \mathbf{x}')\}^T & \mathcal{L}^2\{\Sigma_g(\mathbf{x}, \mathbf{x}')\} \end{bmatrix} \right). \quad (11)$$

We can then write the conditional distribution of the polynomial as $g|\mathcal{L}\{g\}$ using standard Gaussian identities.

3 Conclusion and outlook

The goal of this brief has been to define Bayesian polynomial chaos, and to articulate how it can enhance the engineer's uncertainty quantification toolkit. Our work takes inspiration from Gaussian processes and uses some of its underlying machinery to offer new vistas for what is effectively a Bayesian perspective of polynomial chaos. For the workshop, we will be sharing data and code, with a focus on applications within the remit of experimental aerodynamics and computational fluid dynamics.

References

- [1] D. Xiu and G. E. Karniadakis, "The wiener-asky polynomial chaos for stochastic differential equations," *SIAM journal on scientific computing*, vol. 24, no. 2, pp. 619–644, 2002.

- [2] A. Prabhakar, J. Fisher, and R. Bhattacharya, “Polynomial chaos-based analysis of probabilistic uncertainty in hypersonic flight dynamics,” *Journal of guidance, control, and dynamics*, vol. 33, no. 1, pp. 222–234, 2010.
- [3] J. A. Witteveen, S. Sarkar, and H. Bijl, “Modeling physical uncertainties in dynamic stall induced fluid–structure interaction of turbine blades using arbitrary polynomial chaos,” *Computers & structures*, vol. 85, no. 11-14, pp. 866–878, 2007.
- [4] R. Ghanem and P. D. Spanos, “Polynomial chaos in stochastic finite elements,” 1990.
- [5] M. D. Spiridonakos, E. N. Chatzi, and B. Sudret, “Polynomial chaos expansion models for the monitoring of structures under operational variability,” *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering*, vol. 2, no. 3, p. B4016003, 2016.
- [6] M. P. Pettersson, G. Iaccarino, and J. Nordstrom, “Polynomial chaos methods for hyperbolic partial differential equations,” *Springer Math Eng*, vol. 10, no. 1007, pp. 978–973, 2015.
- [7] P. G. Constantine, M. S. Eldred, and E. T. Phipps, “Sparse pseudospectral approximation method,” *Computer Methods in Applied Mechanics and Engineering*, vol. 229, pp. 1–12, 2012.
- [8] P. R. Conrad and Y. M. Marzouk, “Adaptive smolyak pseudospectral approximations,” *SIAM Journal on Scientific Computing*, vol. 35, no. 6, pp. A2643–A2670, 2013.
- [9] P. Seshadri, A. Narayan, and S. Mahadevan, “Effectively subsampled quadratures for least squares polynomial approximations,” *SIAM/ASA Journal on Uncertainty Quantification*, vol. 5, no. 1, pp. 1003–1023, 2017.
- [10] J. Peng, J. Hampton, and A. Doostan, “A weighted 11-minimization approach for sparse polynomial chaos expansions,” *Journal of Computational Physics*, vol. 267, pp. 92–111, 2014.
- [11] A. Narayan, “Computation of induced orthogonal polynomial distributions,” *arXiv preprint arXiv:1704.08465*, 2017.
- [12] J. M. Hokanson and P. G. Constantine, “Data-driven polynomial ridge approximation using variable projection,” *SIAM Journal on Scientific Computing*, vol. 40, no. 3, pp. A1566–A1589, 2018.
- [13] J. D. Jakeman, F. Franzelin, A. Narayan, M. Eldred, and D. Plfüger, “Polynomial chaos expansions for dependent random variables,” *Computer Methods in Applied Mechanics and Engineering*, vol. 351, pp. 643–666, 2019.
- [14] C. Y. Wong, P. Seshadri, and G. Parks, “Extremum global sensitivity analysis with least squares polynomials and their ridges,” *arXiv preprint arXiv:1907.08113*, 2019.
- [15] G. Geraci, P. M. Congedo, R. Abgrall, and G. Iaccarino, “High-order statistics in global sensitivity analysis: Decomposition and model reduction,” *Computer Methods in Applied Mechanics and Engineering*, vol. 301, pp. 80–115, 2016.
- [16] K. Duraisamy, G. Iaccarino, and H. Xiao, “Turbulence modeling in the age of data,” *Annual Review of Fluid Mechanics*, vol. 51, pp. 357–377, 2019.
- [17] M. Girolami, A. Gregory, G. Yin, and F. Cirak, “The statistical finite element method,” *arXiv preprint arXiv:1905.06391*, 2019.
- [18] D. S. Lubinsky, “A survey of weighted approximation for exponential weights,” *arXiv preprint math/0701099*, 2007.
- [19] S. Rogers and M. Girolami, *A first course in machine learning*. CRC Press, 2016.
- [20] R. C. Smith, *Uncertainty quantification: theory, implementation, and applications*, vol. 12. Siam, 2013.
- [21] P. S. Palar, T. Tsuchiya, and G. T. Parks, “Multi-fidelity non-intrusive polynomial chaos based on regression,” *Computer Methods in Applied Mechanics and Engineering*, vol. 305, pp. 579–606, 2016.
- [22] M. A. Álvarez, L. Rosasco, and N. D. Lawrence, “Kernels for vector-valued functions: A review,” *Foundations and Trends® in Machine Learning*, vol. 4, no. 3, pp. 195–266, 2012.
- [23] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, Í. Polat, Y. Feng,

E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python,” *Nature Methods*, vol. 17, pp. 261–272, 2020.