14th International Conference on Applications of Statistics and Probability in Civil Engineering, ICASP14
Dublin, Ireland, July 9-13, 2023

Constrained non-intrusive polynomial chaos expansion for physics-informed machine learning regression

Himanshu Sharma
Graduate Student, Dept. of Civil and Systems Engineering, Johns Hopkins University, Baltimore, USA

Michael Shields
Professor, Dept. of Civil and Systems Engineering, Johns Hopkins University, Baltimore, USA

Lukas Novak
Professor, Faculty of Civil Engineering, Brno University of Technology, Brno, Czech Republic

ABSTRACT: This work presents a constrained polynomial chaos expansion (PCE) as a physics-informed machine learning (ML) technique to supplement data with physical constraints in the regression framework. PCE is a popular metamodeling technique for uncertainty quantification of expensive computational models. PCE metamodels can also be trained on data in classical ML regression settings to yield pointwise predictions. However, standard PCE metamodel may yield predictions that can violate the underlying physical constraints on the model. In this work, we propose a constrained PCE approach that incorporates the constraints using virtual points in the input domain to solve a constrained least square optimization problem for the PCE coefficients. The resulting constrained PCE model provides an improved fit and leverage from the additional information on the physics of the model. The proposed approach is applied to datasets from 1D analytical functions to impose different types of physical constraints.

1. Introduction

In recent years, there has been a significant increase in the development and application of machine learning models, mainly due to their robustness in identifying trends in complex systems (Jordan and Mitchell (2015)). It is now computationally feasible to train ML models like a neural network over datasets of millions of data points (Dean et al. (2012)). However, in most scientific applications, there is often a paucity of available training data for ML models since performing physical experiments or numerical simulations is typically quite expensive. Furthermore, training a traditional ML model on such datasets often produces predictions that do not agree with the underlying physical constraints, especially in the extrapolatory region. Hence, it is imperative to develop models that can be trained on relatively small datasets with a framework to account for the physics of the system. Such a framework is known as physics-informed machine learning (Karniadakis et al. (2021)).

Researchers have developed frameworks for incorporating physical constraints in existing ML models such as neural networks, Gaussian processes, etc. (Jones et al. (2018); Ling et al. (2016); Swiler et al. (2020)). Such constrained ML models leverage the additional information of physics to supplement limited data and make realistic and improved predictions.

Recently, Torre et al. (2019) presented polynomial chaos expansion (PCE) as a statistical ML regression algorithm. PCE is a well-established metamodeling technique in the field of uncertainty quantification (UQ) (Ghanem and Spanos (1990)). PCE in the UQ setting characterizes the response of a stochastic system as polynomial functions of random inputs. The resulting metamodel is computationally cheaper but accurate in comparison to the original model. Furthermore, the estimates of various output statistics, such as mean, variance, probability density function, sensitivity indices, etc., can be easily computed using the PCE metamodel. In the context of ML, the task is to train the PCE metamodel using available data points to make pointwise predictions. The machine-learned PCE metamodel is constructed by solving for the PCE coefficients in a non-intrusive manner using regression. The accuracy of the PCE metamodel is shown by Torre et al. (2019) to be comparable to existing ML regression models like a neural network, support vector machines, etc.

In this work, we introduce PCE as a data-driven physics-informed machine learning method that incorporates a wide variety of physical constraints. The PCE coefficients are obtained using objective/loss functions, which penalize constraint violation. The constraints are integrated using virtual points in the input domain. Virtual points incorporate the information about the physics of the computational model that the machine-learned PCE metamodel should obey. We use a penalty factor approach (Arora (2004)) to transform the constrained optimization problem into an unconstrained one. A numerical solver based on the quasi-Newton method is used to solve the optimization problem. Further, a heuristic algorithm is proposed to identify the number of virtual points required for a desired fit. Numerical results are obtained using different 1D analytical functions for various commonly encountered physical constraints such as non-negativity, boundedness, monotonicity, convexity, and boundary conditions.

To the best knowledge of the authors, employing PCE in a physics-constrained framework is presented for the first time in this article.

The article is organized as follows. Section 2 provides an overview of PCE in an ML regression setup. In Section 3, the proposed constrained PCE methodology is presented. Numerical results considering various types of physical constraints are obtained using the proposed approach in Section 4. Conclusions are made in the final section.

2. Polynomial chaos expansion in ML regression setting

The task in an ML regression setting is to predict a target variable Y for any d-dimensional predictor variable X, given training set $(\mathscr{X},\mathscr{Y})$. The ML algorithm creates a mapping $\mathscr{M}: X \mapsto Y$ based on available training data $\mathscr{X} = x^{(1)},\ldots,x^{(n)}$ of input observations and of the corresponding output values $\mathscr{Y} = y^{(1)},\ldots,y^{(n)}$, where $y^{(i)} = \mathscr{M}\left(x^{(i)}\right) + \varepsilon$ and ε is a noise term.

In this context, PCE can be employed to construct an analytical model $Y_{PC} = \mathscr{M}_{PC}(X)$, mapping a random input vector to a random output variable Y. Under the assumption of finite variance for Y ($\mathbb{V}(Y) < \infty$), the polynomial chaos representation of Y can be written as,

$$Y(X) = \sum_{\alpha \in \mathbb{N}^d} y_\alpha \Psi_\alpha(X), \tag{1}$$

where $\Psi_\alpha(.)$ are the orthonormal polynomials (e.g., Hermite polynomials for Gaussian random variables, Legendre polynomials for uniform random variables, etc.) (Xiu and Karniadakis (2002)) and $\alpha \in \mathbb{N}^d$ is the multi-index set and each element α_i denotes the degree of $\Psi_\alpha(.)$ in the i^{th} variable, $i = 1,...,d$. The degree of $\Psi_\alpha(.)$ is $|\alpha| = \sum_i \alpha_i$.

We use a standard truncation scheme (Xiu and Karniadakis (2002)) to truncate the infinite series in Eq. (1) as

$$\mathscr{A} = \left\{\alpha \in \mathbb{N}^d : \|\alpha\|_1 \leq p\right\}, \tag{2}$$

where p is the order of the PCE. The total number of elements in $\mathscr{A}$ is given by $\frac{(d+p)!}{d!p!}$.

The objective of PCE regression is to determine the coefficients y_α of the expansion, truncated at some polynomial order, given an initial set $(\mathscr{X},\mathscr{Y})$

of observations (the training set or experimental design). The benefit of using an orthonormal multivariate basis to represent the functional relationship between the inputs and the output lies in its spectrum decay properties which guarantees a rapidly decaying spectrum of the PCE coefficients. Hence, fewer regression coefficients are required for the PCE representation, which avoids overfitting issues. In ML regression settings, the superior performance of the orthonormal basis compared to the non-orthogonal basis is demonstrated on simulated data by Torre et al. (2019).

The PCE coefficients can be computed using regression in a non-intrusive manner by solving the ordinary least squares (OLS) problem. Considering that the cardinality of set $\mathscr{A}$ is smaller than the number of observations n, the OLS problem is given as

$$\begin{aligned} y &= \arg\min_{\tilde{y}} \sum_{j=1}^{n} \left(y^{(j)} - Y_{PC}\left(x^{(j)}\right)\right)^2 \\ &= \arg\min_{\tilde{y}} \sum_{j=1}^{n} \left(y^{(j)} - \sum_{k=1}^{|\mathscr{A}|} y_k \Psi_{\alpha_k}\left(x^{(j)}\right)\right)^2 . \end{aligned} \tag{3}$$

The closed-form solution of Eq. (3) is

$$y = \left(A^T A\right)^{-1} A^T \begin{pmatrix} y^{(1)} \\ \vdots \\ y^{(n)} \end{pmatrix}, \tag{4}$$

where $A_{jk} = \Psi_{\alpha_k}\left(x^{(j)}\right)$, $j = 1,\dots,n$ and $k = 1,\dots,|\mathscr{A}|$. For $n < |\mathscr{A}|$, the solution of Eq. (3) is no longer unique.

There are also sparse techniques to obtain the coefficients like LARS, Lasso, Ridge, etc., which avoids overfitting (Torre et al. (2019); James et al. (2013)). However, in this work, we have used OLS for comparison with the constrained PCE.

3. Constrained polynomial chaos expansion

The objective of this section is to incorporate the physical constraints information of model $\mathscr{M}$ in the PCE regression framework described in Section 2. The difficulty in applying constraints is that it typically calls for a condition to hold globally, which is computationally infeasible. Hence, we approach this problem by relaxing the global requirement and enforcing the constraints only at discrete locations in the input domain, referred to as virtual points. Incorporating this in the OLS framework yields a constrained optimization problem, given as

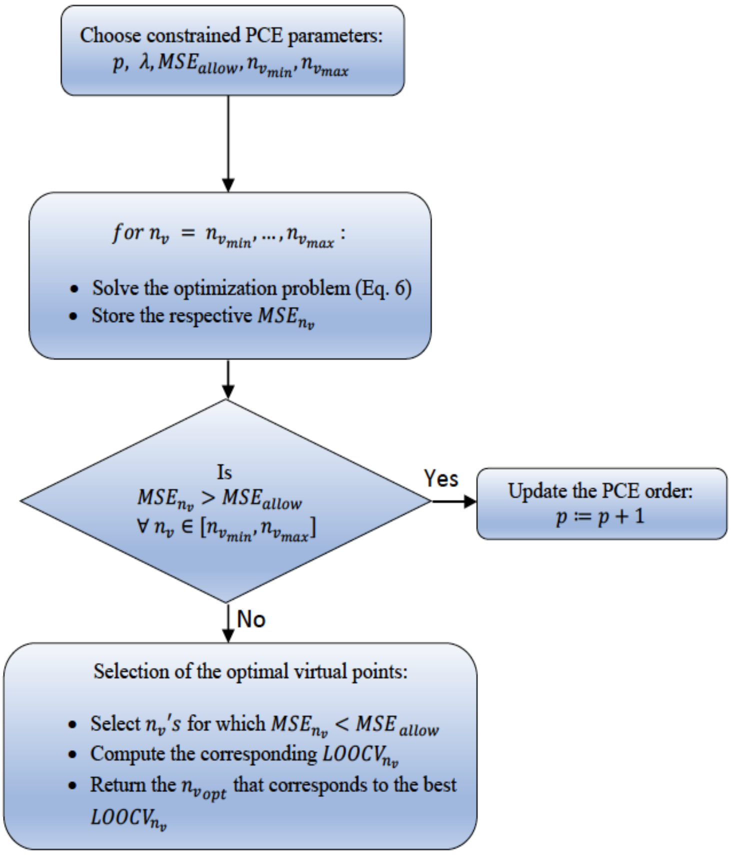

Choose constrained PCE parameters:
$p,\ \lambda, MSE_{allow}, n_{v_{min}}, n_{v_{max}}$
$for\ n_v = n_{v_{min}}, \dots, n_{v_{max}}$:
Solve the optimization problem (Eq. 6)
Store the respective MSE_{n_v}
Is $MSE_{n_v} > MSE_{allow}$ $\forall\, n_v \in [n_{v_{min}}, n_{v_{max}}]$
Yes
Update the PCE order: $p \coloneqq p+1$
No
Selection of the optimal virtual points:
Select n_v's for which $MSE_{n_v} < MSE_{allow}$
Compute the corresponding $LOOCV_{n_v}$
Return the $n_{v_{opt}}$ that corresponds to the best $LOOCV_{n_v}$

Figure 1: Flowchart of the heuristic algorithm

$$y = \arg\min_{\tilde{y}} \sum_{j=1}^{n} \left(y^{(j)} - \sum_{k=1}^{|\mathscr{A}|} y_k \Psi_{\alpha_k}\left(x^{(j)}\right)\right)^2 , \tag{5}$$

subject to

$$\begin{aligned} \text{Non-negativity}: Y_{PC}(x_v^{(i)}) &\geq 0 \quad & i = 1,2,\dots,N_v \\ \text{Boundedness}: a_i \leq Y_{PC}(x_v^{(i)}) &\leq b_i \quad & i = 1,2,\dots,N_v \\ \text{Equality}: Y_{PC}(x_v^{(i)}) &= c_i \quad & i = 1,2,\dots,N_v \\ \text{Monotonocity}: Y'_{PC}(x_v^{(i)}) &\geq 0 \quad & i = 1,2,\dots,N_v \\ \text{Convexity}: Y''_{PC}(x_v^{(i)}) &\geq 0 \quad & i = 1,2,\dots,N_v \end{aligned}$$

where N_v is the number of virtual points.

To solve this constrained optimization problem, we convert (5) into an unconstrained optimization problem using the penalty factor approach, as

$$\begin{aligned} y =& \arg\min_{\tilde{y}} \sum_{j=1}^{n} \left(y^{(j)} - \sum_{k=1}^{|\mathscr{A}|} y_k \Psi_{\alpha_k} \left(x^{(j)} \right) \right)^2 \\ &+ \sum_{\mathrm{i}=1}^{N_v} \lambda_i \left\langle C^{(i)} \right\rangle^2 , \end{aligned} \tag{6}$$

where

$$\left\langle C^{(i)} \right\rangle = \begin{cases} 0 & \text{if constraint is satisfied at } x_v^{(i)} \\ C^{(i)} & \text{if constraint is violated at } x_v^{(i)} \end{cases}$$

and λ_i is the user-defined penalty factor. The number of virtual points regulates the behavior of the polynomials, and thus it is important to find the optimal number of virtual points for a desired fit. We propose a heuristic algorithm to identify the optimal number of virtual points. For a given PCE order p, the algorithm works by incrementally increasing the number of virtual points (n_v) from a predefined $n_{v_{min}}$ (e.g., 30) to $n_{v_{max}}$ (e.g., 60) and store the respective mean square error by solving Eq. (6). Then, it identifies n_v's for which MSE_{n_v} is less than the allowable MSE_{allow}. The n_v that corresponds to the best leave-one-out cross-validation ($LOOCV_{n_v}$) is the optimal equidistant number of virtual points in the input domain D. Another variant of this algorithm based on bootstrapping (i.e., randomly picking a fixed number of virtual points in D) can also be used with a similar approach. Fig. 1 shows the computational flowchart of the heuristic algorithm.

4. NUMERICAL RESULTS

In this section, we present numerical examples to demonstrate the effectiveness of the proposed constrained PCE in handling different types of constraints. We use datasets from 1D analytical functions with known constraints such as non-negativity, boundedness, monotonicity, convexity, and boundary conditions to compare the performance of the constrained PCE with unconstrained PCE. The relative L^2 norm error between the predicted values and true values over a set of test points is used to compare the accuracy and fit of the two approaches. For the examples below, we use $N = 100$ equidistant test points over the input domain D. We solve the optimization problem (Eq. (6)) in Python using the SciPy package (Jones et al. (2001)). For unconstrained PCE, we use the UQpy python package (Olivier et al. (2020)). The optimal number of virtual points is evaluated using the heuristic algorithm shown in Figure 1. For the examples below, we consider the input x as a uniform random variable $x \sim \mathscr{U}([0,1])$, and therefore, Legendre orthonormal polynomials are used as PCE basis functions (Xiu and Karniadakis (2002)).

4.1. *Example 1*

Consider the following non-negative function:

$$f(x) = \frac{1}{100} + \frac{5}{8}(2x-1)^4 \left[(2x-1)^2 + 4\sin(5\pi x)^2 \right], \tag{7}$$

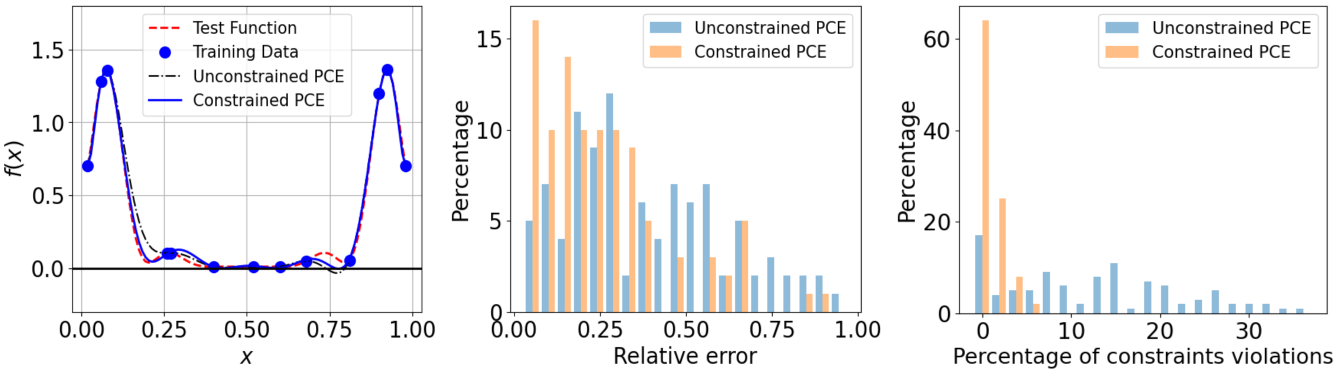
where $x \in [0,1]$. We consider $p = 12$ and a dataset with $n = 13$ training points. Figure 2(a) shows the comparison of the relative fit of constrained and unconstrained PCE with the original function. It can be seen that the constrained PCE provides a better fit while satisfying the underlying non-negativity constraint. To demonstrate the robustness of the proposed method, we train the model using 100 uniformly distributed datasets (simulations). Figure 2(b) shows the histogram of the relative L^2 error for different datasets. As can be observed, the histogram for the constrained PCE is more heavily weighted towards the lower relative error than its counterpart, demonstrating the superior performance of the proposed method. Figure 2(c) shows the histogram of the percentage of constraint violations at test points for different datasets considering both constrained and unconstrained PCE. It is evident that the unconstrained PCE violates the non-negativity constraints for many datasets, while the constrained PCE satisfies the constraints very well.

4.2. *Example 2*

Consider the following monotonic function:

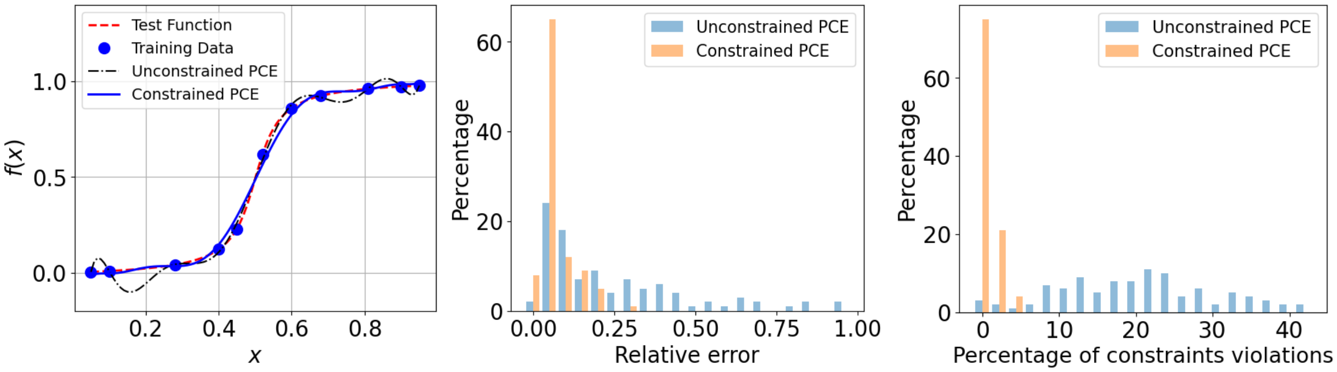
$$f(x) = \frac{1}{3} \left[\tan^{-1}(20x-10) - \tan^{-1}(-10) \right], \tag{8}$$

where $x \in [0,1]$. We use $p = 9$ and $n = 11$ training points. Figure 3(a) shows the relative fit of



(a) *(b)* *(c)*

Figure 2: Non-negativity constraint: (a) Comparison of constrained PCE and unconstrained PCE with the test function. (b) Histogram of relative L^2 error of 100 training sets for constrained and unconstrained PCE. (c) Histogram of percent constraint violations for 100 training sets for constrained and unconstrained PCE.



(a) *(b)* *(c)*

Figure 3: Monotonicity constraint: (a) Comparison of constrained PCE and unconstrained PCE with the test function. (b) Histogram of relative L^2 error of 100 training sets for constrained and unconstrained PCE. (c) Histogram of percent constraint violations for 100 training sets for constrained and unconstrained PCE.

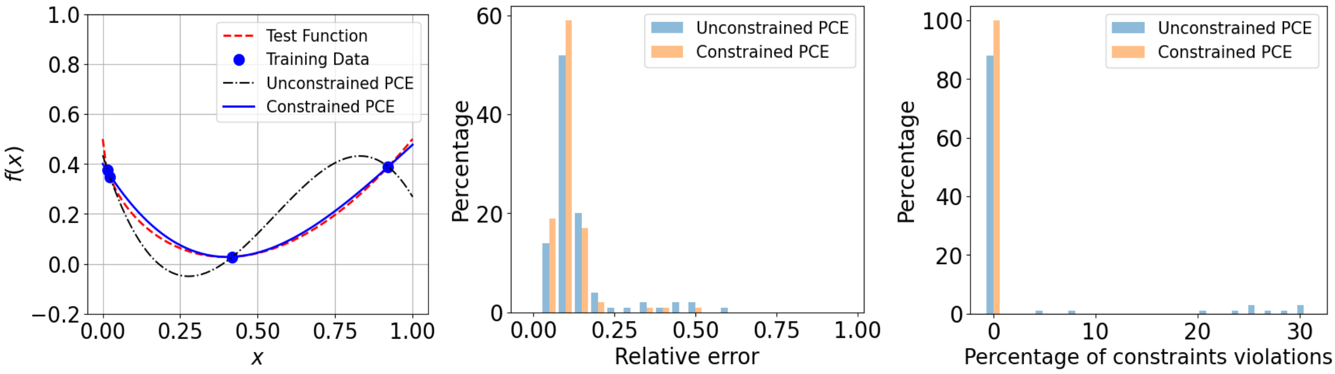
the unconstrained and constrained PCE compared to the given monotonic function. It can be observed that the constrained PCE satisfies the monotonicity constraint and provides a better fit compared to the unconstrained PCE. Again, to show the algorithm's robustness in handling the monotonicity constraint, we train the constrained PCE model for different datasets uniformly distributed and report the relative L^2 error. Figure 3(b) compares the relative error of the two methods, and it can be observed that the constrained PCE results in a smaller relative error compared to the unconstrained PCE, which demonstrates an improved fit. Figure 3(c) shows a histogram of constraint violations at test points for different datasets, indicating that constrained PCE preserves the monotonicity constraints for most datasets compared to unconstrained PCE.

4.3. *Example 3*

Consider the following convex function:

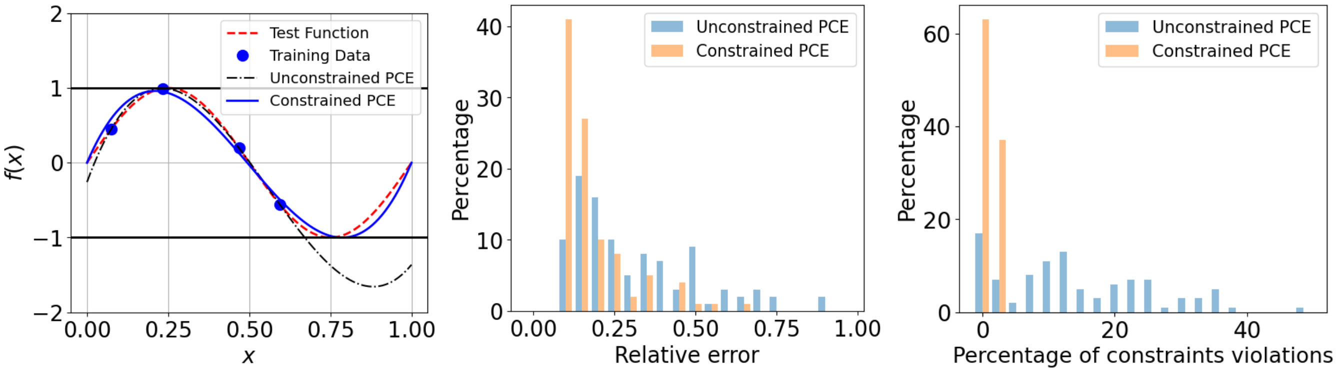
$$f(x) = x^2 - \sqrt{x} + 0.5, \quad x \in [0,1]. \qquad (9)$$

For this example, we use $p = 3$ and $n = 4$ training points. Figure 4(a) shows the plot of the given convex function, comparing it with the predicted curves from the proposed constrained method and its counterpart. Again, we can observe an improved fit while maintaining the inherent constraint for constrained PCE. The relative error and constraint violations for different datasets are shown in



(a) *(b)* *(c)*

Figure 4: Convexity constraint: (a) Comparison of constrained PCE and unconstrained PCE with the test function. (b) Histogram of relative L^2 error of 100 training sets for constrained and unconstrained PCE. (c) Histogram of percent constraint violations for 100 training sets for constrained and unconstrained PCE.



(a) *(b)* *(c)*

Figure 5: Boundedness and boundary conditions: (a) Comparison of constrained PCE and unconstrained PCE with the test function. (b) Histogram of relative L^2 error of 100 training sets for constrained and unconstrained PCE. (c) Histogram of percent constraint violations for 100 training sets for constrained and unconstrained PCE.

Figure 4(b) and Figure 4(c), respectively. As expected, the relative error and constraint violations are small for constrained PCE, demonstrating superior performance.

4.4. Example 4

$$f(x) = \sin 2\pi x, \quad x \in [0,1] \tag{10}$$

For this example, we use $p = 3$ and $n = 4$. We impose two types of constraints: Boundedness ($-1 \leq f(x) \leq 1$) and boundary conditions ($f(x) = 0$ @ $x = 0,\ 1$). Figure 5(a) shows the comparison of constrained and unconstrained PCE fit with the original function. Again, we can observe an improved fit for the proposed method while preserving the constraints. Figures 5(b) and 5(c) show the histogram of relative error and constraint violations, respectively, for different datasets. Again, we observed that the performance of constrained PCE is superior to that of unconstrained PCE.

5. CONCLUSIONS

A new constrained polynomial chaos expansion method incorporating physical constraints in the PCE regression framework is presented. The constraints are integrated into the OLS regression using virtual points in the input domain, yielding a constrained optimization problem. A heuristic algorithm is proposed to find the optimal number of virtual points required for a desired fit. The performance of the proposed method is compared with unconstrained PCE using different 1D analytical functions with constraints like non-negativity, boundedness, monotonicity, convexity, and boundary conditions. Numerical results indicate that the proposed method provides an improved fit to the training data while preserving the underlying constraints, which makes it suitable for scientific applications. However, the curse of dimensionality associated with PCE makes it imperative to develop a sparse implementation using the proposed approach, which will be the subject of further work.

6. REFERENCES

Arora, J. (2004). *Introduction to optimum design*. Elsevier.

Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., Ranzato, M., Senior, A., Tucker, P., Yang, K., et al. (2012). "Large scale distributed deep networks." *Advances in neural information processing systems*, 25.

Ghanem, R. and Spanos, P. D. (1990). "Polynomial chaos in stochastic finite elements.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An introduction to statistical learning*, Vol. 112. Springer.

Jones, E., Oliphant, T., Peterson, P., et al. (2001). "Scipy: Open source scientific tools for python.

Jones, R. E., Templeton, J. A., Sanders, C. M., and Ostien, J. T. (2018). "Machine learning models of plastic flow based on representation theory." *arXiv preprint arXiv:1809.00267*.

Jordan, M. I. and Mitchell, T. M. (2015). "Machine learning: Trends, perspectives, and prospects." *Science*, 349(6245), 255–260.

Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., and Yang, L. (2021). "Physics-informed machine learning." *Nature Reviews Physics*, 3(6), 422–440.

Ling, J., Jones, R., and Templeton, J. (2016). "Machine learning strategies for systems with invariance properties." *Journal of Computational Physics*, 318, 22–35.

Olivier, A., Giovanis, D. G., Aakash, B., Chauhan, M., Vandanapu, L., and Shields, M. D. (2020). "Uqpy: A general purpose python package and development environment for uncertainty quantification." *Journal of Computational Science*, 47, 101204.

Swiler, L. P., Gulian, M., Frankel, A. L., Safta, C., and Jakeman, J. D. (2020). "A survey of constrained gaussian process regression: Approaches and implementation challenges." *Journal of Machine Learning for Modeling and Computing*, 1(2).

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

Xiu, D. and Karniadakis, G. E. (2002). "The wiener–askey polynomial chaos for stochastic differential equations." *SIAM journal on scientific computing*, 24(2), 619–644.