International Journal for Uncertainty Quantification, 2 (2): 125–143 (2012)

FORWARD AND BACKWARD UNCERTAINTY PROPAGATION FOR DISCONTINUOUS SYSTEM RESPONSE USING THE PADÉ-LEGENDRE METHOD

***Tonkid Chantrasmi*[1] *& Gianluca Iaccarino*[2,*]**

[1] *Department of Mechanical and Aerospace Engineering, King Mongkut's University of Technology North Bangkok, Bangkok 10800, Thailand*

[2] *Mechanical Engineering Department, Stanford University, Stanford, California, 94305, USA*

Original Manuscript Submitted: 05/20/2011; Final Draft Received: 09/17/2011

The Padé-Legendre method has been introduced as an effective approach to characterize uncertainties in the presence of strongly non-linear or discontinuous system responses—thus, it supports forward propagation. The method is based on the construction of a ratio of polynomials that approximate the available data. Two criteria for the choice of the best approximant are considered and an optimization approach is proposed. Moreover, the approach is applied in a case in which the discontinuity in the system response is due to limited data, to demonstrate how the successive addition of data transforms the rational approximant into a simple polynomial interpolant (the denominator becomes a constant). Finally, the present method is applied to estimate an input parameter characterized by a sharp discontinuity, using Bayesian inference starting from observations of the system response—thus, it also supports backward propagation.

KEY WORDS: *uncertainty quantification, Padé-Legendre reconstruction, discontinuity, Bayesian inference*

1. INTRODUCTION

In many physical problems governed by non-linear mathematical models, a discontinuous behavior of the output of interest is observed in response to a smooth variation of the system inputs. The analysis of this behavior is particularly important in situations in which variability in the output has to be characterized statistically. In recent years, uncertainty quantification (UQ) has gained popularity as a methodology to assess the effect of variability and lack of knowledge on the output of a computational model. This problem is referred to as a forward uncertainty propagation problem, and contrasted to situations in which starting from observed system performance (output) one infers the input quantities (backward uncertainty propagation). Probabilistic methodologies based on sampling can be readily applied by converting the source of uncertainties (boundary conditions, model parameters, etc.) into random variables or fields. This approach leads to a comprehensive statistical characterization of the outputs but suffers from a slow convergence that makes its application to realistic problems impractical. More recently, the use of a polynomial basis to represent the dependency of the solution on the uncertain inputs has gained popularity [1–3] because of its performance in computing statistical moments. The main reason behind the effectiveness of polynomial-based approaches is the underlying assumption that the system response is smooth and can be approximated accurately with low-order polynomial expansions. In strongly non-linear problems, such as compressible fluid dynamics, multi-material heat transfer, multi-phase flows, etc., the potential lack of smoothness can lead to an inaccurate polynomial reconstruction of the system response because of the occurrence of Gibbs oscillations. In this paper, we explore a recently developed technique for handling discontinuous responses, the Padé-Legendre (PL) approach [4, 5]; the PL method uses a ratio of polynomial expansions to reconstruct the surface response and allows one to represent genuinely discontinuous

*Correspond to Gianluca Iaccarino, E-mail: jops@stanford.edu, URL: http://uq.stanford.edu/

2152–5080/12/$35.00

functions without oscillations. Two main extensions are considered in this paper, the first is related to the use of an optimization approach to define the parameters in the PL reconstructions (as multiple choices are available with a given set of data). The proposed automatic parameter selection is shown to be effective in multi-scale situations in which a discontinuity is the result of limited data (or a coarse grid representation), and the successive addition of data leads to a smooth response. In this case the algorithm correctly reverts to a pure polynomial reconstruction (the denominator becomes a constant). The second contribution is the formulation of the PL approach within a Bayesian inference procedure. The presence of a discontinuity in the input can lead to a purely posterior condition and inhibit the convergence of the inversion procedure.

1.1 Stochastic Collocation Method

In recent years, two alternative approaches to Monte Carlo simulations have found relatively widespread use in probabilistic UQ: stochastic Galerkin [1, 6–9] and stochastic collocation [10–13]. Stochastic Galerkin approaches are based on a representation of the uncertain solution as a functional expansion in terms of polynomials. These schemes are intrusive, in the sense that the deterministic solvers are modified to incorporate the stochastic expansions. On the other hand, in the stochastic collocation method the deterministic solver is used unmodified for simulations at sets of input values, typically corresponding to quadrature points, resulting in a non-intrusive approach.

Mathematically, we write the output, u, as a linear combination of the orthogonal basis polynomials, Φ_i of the random input, ξ:

$$u(x,\xi) = \sum_{i=0}^{\infty} \hat{u}_i(x)\Phi_i(\xi), \tag{1}$$

where x is the physical coordinate and $\hat{u}_i$ are the coefficients to be determined. The summation in Eq. (1) is truncated at a finite $N \in \mathbb{N}$, so that the coefficients can be computed from available data. This results in a projection of the real solution u into the space spanned by Φ_0, Φ_1, ... Φ_N:

$$u_N(x,\xi) = \sum_{i=0}^{N} \hat{u}_i(x)\Phi_i(\xi). \tag{2}$$

To calculate the coefficients $\hat{u}_i$, we utilize the orthogonality properties of Φ_i. Taking discrete scalar product with Φ_k for $k = 0, 1, ..., N$, we get uncoupled equations for $\hat{u}_k$:

$$\hat{u_k} = \frac{\langle u, \Phi_k \rangle_N}{\langle \Phi_k, \Phi_k \rangle_N}. \tag{3}$$

The scalar product is defined as

$$\langle \phi, \psi \rangle_N = \sum_{j=0}^{N} \phi(\xi_j)\psi(\xi_j)w_j, \tag{4}$$

where the quadrature points ξ_j and the associated weights w_j are predefined for ξ_j characterized by standard probability distributions [14].

The algorithm to compute the approximation of u is as follows. First, perform deterministic calculations at the predefined collocation points ξ_j. Next, calculate $\hat{u_k}$ from Eq. (3). Than, plug the coefficients into Eq. (2) to obtain the expression for the approximation u_N as a function of the uncertain input, ξ. With this expression, one can now efficiently sample a large number of (approximated) solutions according to the distribution of ξ to generate random realizations of solution u or compute its statistics. In the following, we will compare this method [referred as stochastic collocation (SC)] to the present Padé-Legendre approach.

The simplest way is to generalize discrete scalar product (4) to operate on a tensor grid and use the tensor product of the one-dimensional polynomial basis. One can alternatively use a sparse grid instead of the tensor grid in high dimensions to alleviate the curse of dimensionality—the required data grow exponentially with respect to the number of uncertain variables [11, 12, 15, 16].

1.2 Padé-Legendre Approximant

In this section, we introduce the PL method in multi-dimensional settings. For the sake of simplicity, the approximation is formulated in a two-dimensional problem. In addition, we will consider only the isotropic cases; i.e., we consider the same number of data points in each direction on a tensor grid. Let us assume that we have $(N+1)\times(N+1)$ data points. The focus on this section is on the algorithm to compute the PL approximant. Readers who are interested in a more detailed discussion of the overall approach are referred to our previous work [4].

In the PL method, we represent the approximation of solution u as a rational function of the uncertain variables. We construct the PL response surface on the combination of physical and stochastic spaces. Denoting the PL approximation as $R(u)$, we write

$$R(u)(x,y) = \frac{P(x,y)}{Q(x,y)} = \frac{\sum_{i=0}^{M} \hat{p}_i \Phi_i(x,y)}{\sum_{i=0}^{L} \hat{q}_i \Phi_i(x,y)}, \tag{5}$$

where M and L are the orders of the expansions at the numerator and denominator, respectively. We use Legendre polynomials for basis Φ, although other bases are also possible. Here, x and y are either physical or stochastic variables; in the latter case we assume their probability distribution to be uniform in a known interval.

We construct $R(u)$ to be a good approximation to u. This is done by minimizing the linear PL approximation error:

$$v_i = \langle P - Qu, \Phi_i \rangle_N, \tag{6}$$

for all two-dimensional polynomial basis Φ_i of total degree at most N. The discrete scalar product is defined as in Eq. (4). In multi-dimensional settings, it is generally impossible to enforce that v_i vanishes for all Φ_i. In our proposed method, we require that $v_i = 0$ only for all polynomial basis of total degree at most M and that v_i is minimized in a least-squares sense for polynomial basis of total degree from $M+1$ to $M+K$ for some positive integer K. This basis of total degree from $M+1$ to $M+K$ is used to calculate denominator Q to ensure that product uQ is smooth. When uQ is smooth, it can be well approximated by P, which is a polynomial of degree at most M by the standard stochastic collocation approach.

Before we proceed to describe the algorithm to calculate $\hat{p}$ and $\hat{q}$, let us note that we have four parameters for construction of the PL approximant—N, M, L, and K. Figure 1 shows the relationships among these parameters. In our previous work [4], the sensitivity of the approximation with respect to these parameters was investigated; in the next section we propose an automatic selection algorithm based on the desired properties of the approximant.

In order to calculate the coefficients in Eq. (5), we first calculate $\hat{q}_i$ from the following system of equations:

$$\begin{bmatrix} \langle u\Phi_1, \Phi_{c(M)+1}\rangle_N & \cdots & \langle u\Phi_{c(L)}, \Phi_{c(M)+1}\rangle_N \\ \vdots & \ddots & \vdots \\ \langle u\Phi_1, \Phi_{c(M+K)}\rangle_N & \cdots & \langle u\Phi_{c(L)}, \Phi_{c(M+K)}\rangle_N \end{bmatrix} \begin{bmatrix} \hat{q}_1 \\ \vdots \\ \hat{q}_{c(L)} \end{bmatrix} = \underline{0}. \tag{7}$$

The system of Eq. (7) is the result of substituting the spectral expansions of P and Q into Eq. (6) and requiring error v_i to be zero for $i = c(M)+1, \ldots, c(M+K)$.

The matrix-vector product on the right-hand side of Eq. (7) is a column vector of v_i for the polynomial basis of total degree from $M+1$ to $M+K$. This system of equations is over-constrained given that $c(M+K)-c(M) > c(L)$ where $c(s) = (s+1)(s+2)/2$. We can obtain the optimal solution in the least-squares sense by using the singular value decomposition of the matrix on the left-hand side of Eq. (7) [17]. This gives $\hat{q}_i$, which allows us to evaluate denominator Q. Once Q is known, the computation of numerator coefficients $\hat{p}_i$ is similar to calculating collocation coefficients for Qu:

$$\hat{p}_n = \frac{\langle P, \Phi_i\rangle_N}{\langle \Phi_i, \Phi_i\rangle_N} = \frac{\langle Qu, \Phi_i\rangle_N}{\langle \Phi_i, \Phi_i\rangle_N} \qquad i = 1, 2, ..., c(M). \tag{8}$$

We now have $\hat{q}_i$ from Eq. (7) and $\hat{p}_i$ from Eq. (8) to plug into Eq. (5), thus obtaining the PL approximant for u.

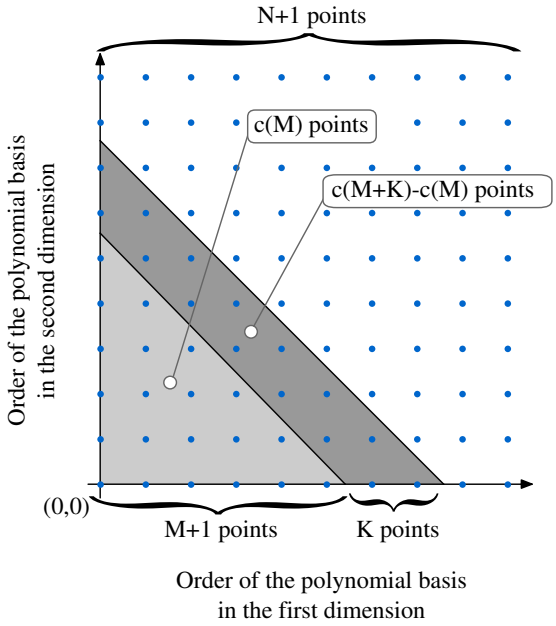


FIG. 1: Schematic representation of the parameters introduced in the Padé-Legendre surface construction and their relationship. The parameter L is not presented in the diagram but has to satisfy the inequality $c(M+K)-c(M) > c(L)$ where $c(s) = (s+1)(s+2)/2$. The light grey area is where v_i vanishes and the dark grey area is where v_i is minimized in least-squares sense.

2. EXTENSIONS AND REFINEMENT OF THE PL APPROACH

In this section, we describe three topics that extend the PL methods: (1) automatic parameter selection (APS), (2) convergence for smooth functions, and (3) the PL method in the inversion problem.

2.1 Automatic Parameter Selection (APS)

The formulation of the Padé-Legendre approximation requires specification of the Padé parameters—K, M, and L. The choice of parameter N is usually dictated by existing data and/or available computational resources. In many cases, a priori knowledge of the underlying function can help an experienced user make a good choice, but in many cases it is difficult to do so without trial and error. An algorithm to automatically select these parameters is a key component of the success of the approach.

In order to compare whether one solution obtained using a parameter set is better than another, one needs a metric. The algorithm proposed here involves two different metrics: the traditional L_2 approximation error and a smoothness indicator based on the concept of total variation (TV).

2.1.1 L_2-Norm Error Estimate

In the present approach, the L_2-norm error estimate is simply the weighted L_2-norm of the difference between the data and the approximated solution. The weights are derived from the quadrature rule since the data are not uniformly distributed. The difference is only taken at the data point to avoid further interpolation. Mathematically, the L_2-norm error estimate is

$$E_{L_2}^2 = \|\tilde{u} - u\|_{L_2}^2 = \sum_{j=1}^{N_q} w_j [\tilde{u}(x_j) - u(x_j)]^2, \tag{9}$$

where N_q, x_j, and w_j are defined in multidimensional settings [2], u is the given data, and $\tilde{u}$ is the approximated solution. Further, we normalize this error estimate by the L_2-norm of the data,

$$\|u\|_{L_2}^2 = \sum_{j=1}^{N_q} w_j u(x_j)^2, \tag{10}$$

resulting in the final expression for the normalized L_2-norm error estimate

$$e_{L_2}^2 = \frac{\|\tilde{u} - u\|_{L_2}^2}{\|u\|_{L_2}^2} = \frac{\sum_{j=1}^{N_q} w_j (u(x_j) - \tilde{u}(x_j))^2}{\sum_{j=1}^{N_q} w_j u(x_j)^2}. \tag{11}$$

Note that the above error estimate is scaled, although it does not have an upper bound. From now on, we refer to the normalized version of the error estimate (11) as the L_2-error estimate.

2.1.2 Smoothness Indicator

The L_2-error estimate is an indicator of how well the approximation interpolates the data points; however, it does not provide information about the approximation in between the data points. In fact, when a discontinuity is present, a large error occurs between data points due to the Gibbs phenomena. A smoothness indicator (SI) is designed to detect artificial oscillation between data points.

The smoothness indicator here is based on the TV concept. In one dimension, the total variation of a function f over an interval $[a, b]$ is defined as the following:

$$TV(f, [a, b]) = \sup_P \sum_{i=0}^{n_P - 1} |f(x_{i+1}) - f(x_i)|, \tag{12}$$

where the supremum is over the set of all possible partitions $P = \{x_0, x_1, \ldots, x_{n_P}\}$ of the interval $[a, b] = [x_0, x_{n_P}]$.

Note that the smallest partitions that would give the (largest) TV value are when $x_1, \ldots, x_{n_P - 1}$ are the locations of the local extrema of f. Adding a point that is not a local extrema to this partition does not change the value of the summation in Eq. (12). Therefore, if f is known at a finely resolved grid in x, the TV can be accurately estimated as the sum of the differences of f between the pairs of all adjacent points; i.e.,

$$TV(f, [a, b]) \approx \sum_{i=1}^{N_g - 1} |f(x_{i+1}) - f(x_i)|, \tag{13}$$

for large N_g and $x_1 < x_2 < \cdots < x_{N_g}$ are nodes on a fine grid for interval $[a, b]$.

In multi-dimensional settings, we simply apply the one-dimensional formula (13) in different dimensions and locations. For example, in three dimensions, assume that the grid (x, y, z) is of size $N_1 \times N_2 \times N_3$, then we apply Eq. (13) in the first dimension $N_2 \times N_3$ times, each along the same x but at different (y, z). Similarly, we perform the one-dimensional TV calculation $N_1 \times N_3$ times in the second dimension and $N_1 \times N_2$ in the third dimension. Overall, for d-dimensional problems, we apply one-dimensional formula (13) a total of

$$N_{\text{applications}} = N_g \times \left(\frac{1}{N_1} + \frac{1}{N_2} + \cdots + \frac{1}{N_d} \right) \tag{14}$$

times, where $N_g = \prod_{i=1}^{i=d} N_i$ is the total number of the grid nodes. For an isotropic grid in d dimensions, with N_1 nodes in each direction, this number becomes $d \times N_1^{d-1}$.

Finally, our smoothness indicator is the sum of all one-dimensional TV indicators from Eq. (13) divided by the total number of applications [Eq. (14)]. For compactness, we write this as

$$E_{\text{SI}} = \text{SI}(\tilde{u}, G_F), \tag{15}$$

where $\tilde{u}$ is the approximated solution given on fine grid G_F. Further, we normalize the smoothness indicator by the data as follows:

$$e_{\mathrm{SI}} = \frac{|\mathrm{SI}(\tilde{u}, G_F) - \mathrm{SI}(u, G_D)|}{\mathrm{SI}(u, G_D)}, \tag{16}$$

where u are the data given on the data grid G_D. Note that, similar to the L_2-error estimate, the smoothness indicator is non-dimensionalized and does not have an upper bound. If the approximated solution does not produce additional extrema from what the data indicate, its smoothness indicator will be zero.

2.1.3 Using Two Metrics

We have defined two different metrics for each choice of the PL parameters. Some cases can be resolved easily; e.g., those with both metrics higher than another one (or one equal and one higher). The remaining parameter sets correspond to possible candidates, or, in optimization context, to a Pareto front.

Let $\mathcal{P}$ be this Pareto front with N_s elements. We can sort the elements of $\mathcal{P}$ based on one error estimate, say e_{L_2}. It is easy to show that this same ordering is also sorted according to the other error estimate, e_{SI}, but in the reverse order. With this ordering, write $\mathcal{P} = \{P_1, P_2, \ldots, P_{N_s}\}$ where $e_{L_2}(P_1) \leq e_{L_2}(P_2) \leq \ldots \leq e_{L_2}(P_{N_s})$ and $e_{SI}(P_1) \geq e_{SI}(P_2) \geq \ldots \geq e_{SI}(P_{N_s})$.

Any choice in $\mathcal{P}$ is logically acceptable. The selection process presented hereafter is heuristic. Ideally, the best choice should be based on its intended application. However, on the other hand, we would like to provide a readily usable choice. To this end, we provide the users one initial choice and give them an option to seek other possible choices as needed.

Our default choice is P_1—most accurate but least smooth—based on the assumption that the smoothness requirement varies from one application to another. In many applications, it is sufficient to achieve a certain minimum level of smoothness. On the other hand, the L_2-norm is a more universal measure and the users generally want to achieve the highest accuracy possible with respect to this metric. According to this argument, we also leave the possibility to request for smoother and smoother solutions while lowering the accuracy with respect to the data. In this manner, if the users require smoother solutions for their applications, the parameter selection algorithm would yield $P_2, P_3, \ldots$, in order.

2.1.4 Example 1: Step Function

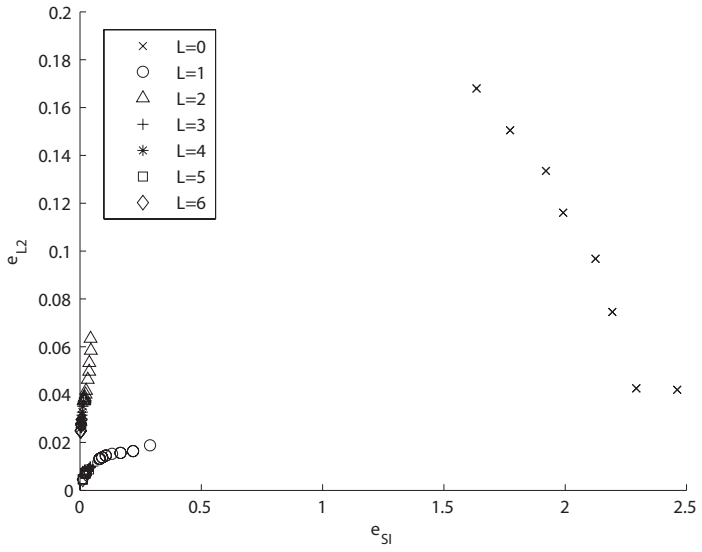
Consider a step function $f(x) = \mathrm{sign}(x)$ where $x \in [-1, 1]$. Let the number of given data points $N = 30$ and the data are given on Legendre-Gauss-Lobatto (LGL) nodes [14]. Here, we illustrate the use of the APS algorithm.

Figures 2(a) and 2(b), shows all possible PL solutions up to $L = 6$ in terms of the two metrics, e_{SI} and e_{L2}. Note that the SC solutions ($L = 0$) have much higher error estimates in this case.[1] Excluding the SC solutions, we see that there are two distinct trends: one for odd L and another for even L. Roughly speaking, we observe that the odd-L solutions have lower e_{L2} but higher e_{SI}, i.e., they are more accurate in the L_2 sense but less smooth.

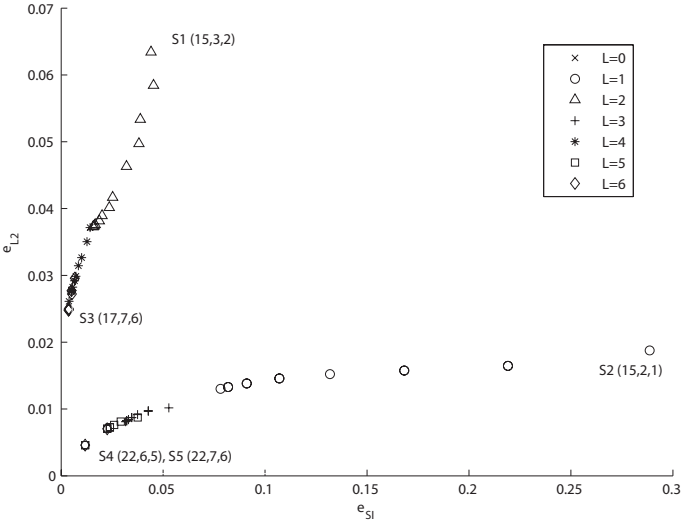
In Fig. 2(b), we define five particular parameter sets, S1-S5. Figures 3(a)–3(d) show these solutions as functions of x. First, we compare S1 and S2 and observe that S1 is smoother. This is immediately clear from the fact that the S2 solution contains large undershoots near the discontinuity. However, the S2 solution is more accurate than S1. The L_2-error in the S1 solution mainly originates at the points closest to the discontinuity, while the S2 solution interpolates these points, and the error is generated as a result of a slight overshoot/undershoot further away.

Now, consider solutions S3, S4, and S5. These three solutions compose the Pareto front of this problem. It turns out that S4 and S5 coincide perfectly, so we only need to compare S3 and S4. Again, we need a trade-off between accuracy and smoothness as in the comparison between S1 and S2. The differences in the errors are much smaller in this case, although they are still noticeable. Solution S3 is smooth and has an L_2-error near the discontinuity; we can visually see this in Fig. 3(c). On the other hand, solution S4 visually passes through all the data points but also

[1] As mentioned earlier, the SC method used here is based on the pseudo-spectral formulation in [18]. In this formulation, due to aliasing error the solution is not necessarily an interpolation of the given data. Thus, the L_2-error estimate is not exactly zero.



(a)



(b)

FIG. 2: All possible PL solutions up to $L = 6$ according to their $e_{\rm SI}$ and e_{L2} measures. (b) is a zoom-in of (a). Five solutions are labeled as S1-S5 in (b) for further references (note: S4 and S5 coincide). The parameters in parentheses are the PL parameters (M, K, L).

contains slightly larger undershoots than those in solution S3. The undershoots for both solutions are too small to observe visually in Fig. 3.

2.1.5 Example 2: Square Wave

Consider a slightly more complicated function: $f(x) = \text{sign}(x + 0.2) - \text{sign}(x - 0.5)$ where $x \in [-1, 1]$ and f is a single square wave in the domain. The square wave spans the interval $[-0.2, 0.5]$ with a height of 2. We will consider two data sets, $N = 20$ and $N = 40$, both given at standard LGL points.

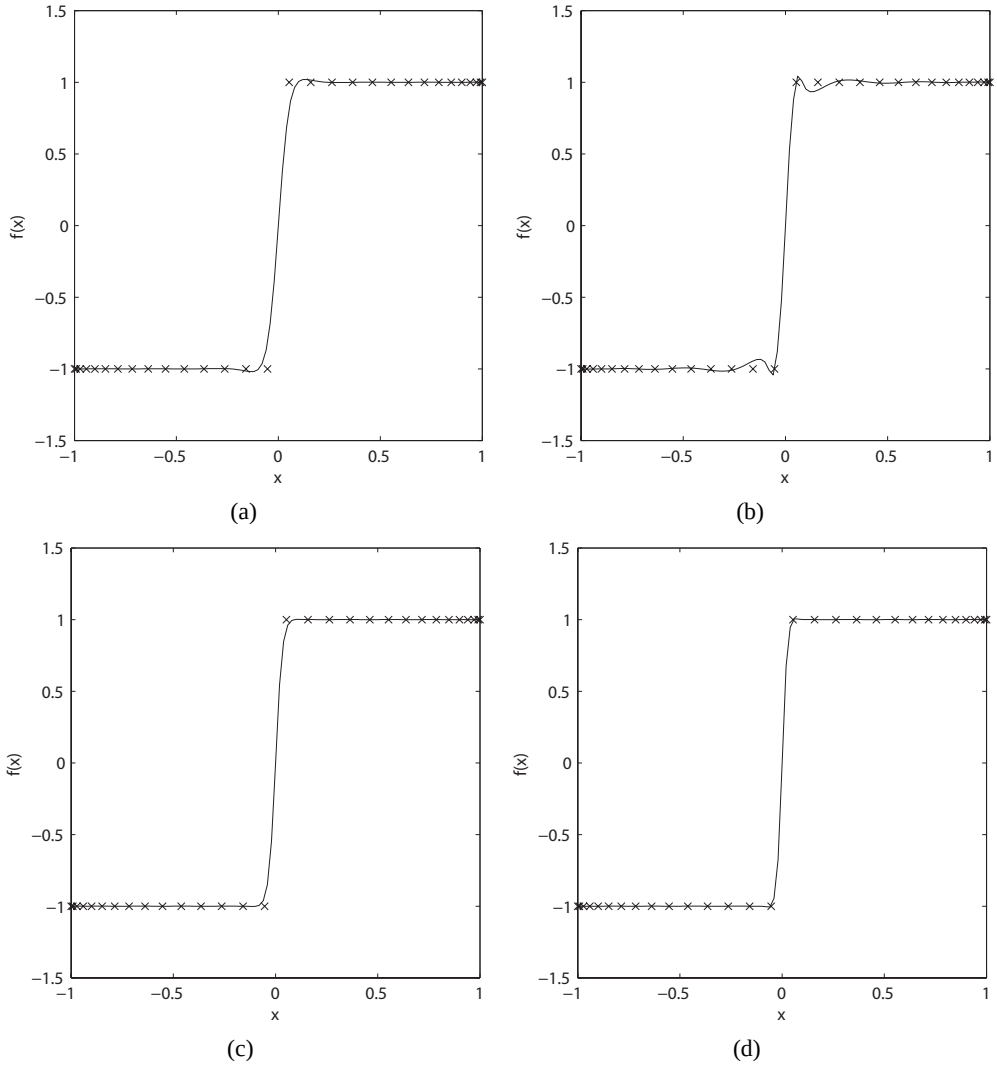


FIG. 3: PL solutions (a) S1, (b) S2, (c) S3, and (d) S4 and S5 as defined in Fig. 2(b).

First, consider the case where $N = 20$. Figure 4 shows all the PL solutions in the Pareto front with their corresponding two metrics. Unlike in the earlier example of a step function, the SC solution is part of the Pareto front here due to its low e_{L2}; however, it does have a relatively high e_{SI}. Upon closer inspection, we found that this SC solution (P_1) is highly oscillatory as seen in Fig. 5(a). In many applications these spurious oscillations are very undesirable; for example, when one wants to find extrema values or detects regions of steep gradient. In such cases, users should request the smoother PL solutions. The next solution P_2 in the Pareto front has smaller e_{SI} than that of P_1 by more than an order of magnitude. This is usually an indication that the present Gibbs oscillation has been effectively suppressed. Other solutions beyond P_2 are smoother but less accurate as expected; however, the solutions only change slightly. Figures 5(b) and 5(c), show P_2 and P_5, respectively.

Next, we consider the same underlying square wave function but with more data, $N = 40$. Figure 6 reports the Pareto front of the PL solutions for this problem. Again, the SC solution is part of the Pareto front as P_1. Here, P_2 has more than an order of magnitude smaller e_{SI} than that of P_1, suggesting that the Gibbs oscillation has been suppressed. The PL solutions beyond P_2 change only gradually as seen in Figures 7(a)–7(d). At closer observation, we see that the solutions P_4 and P_6 do not quite pass through the data points near the discontinuities in contrast to P_2. However,

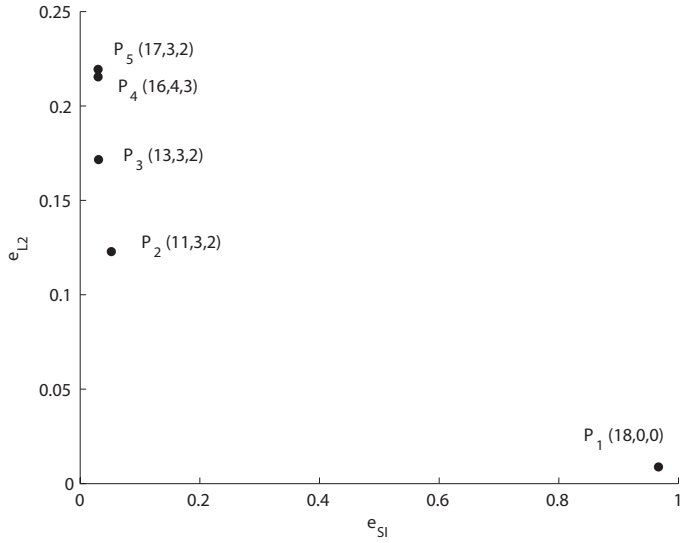
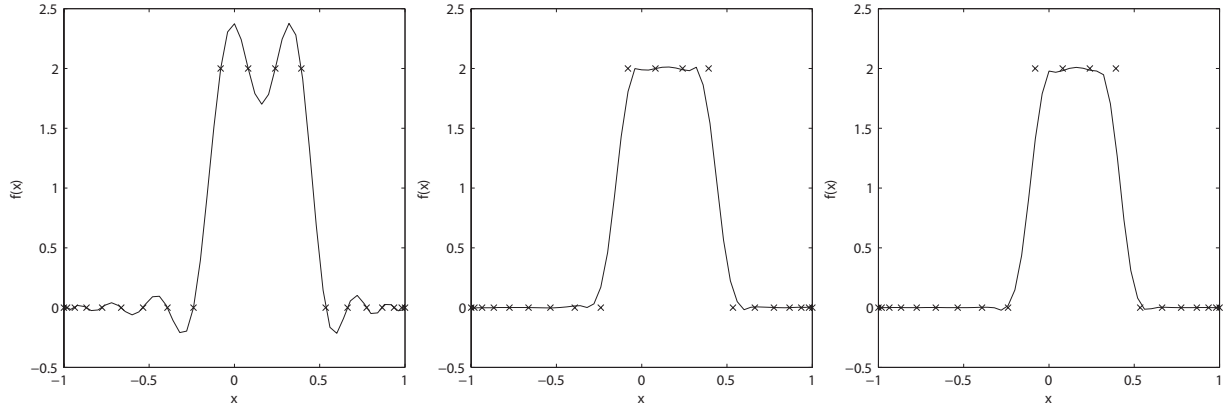


FIG. 4: The Pareto front of PL solutions of the square wave problem with $N = 20$.



(a) (b) (c)

FIG. 5: Some of the PL solutions in the Pareto front for the square wave problem with $N = 20$. (a) P_1 (most accurate), (b) P_2, and (c) P_5 (smoothest).

the solution P_2 has small undershoots near the discontinuities. This is the same behavior observed in the step function example. We suspect that this behavior is not observed in the case of $N = 20$, because there are not enough points separating the two discontinuities.

2.2 Convergence for Continuous Functions

In Section 2.1, we defined smoothness indicator (16) of the approximated solution with respect to the given data. A more subtle question to ask is: How smooth are the given data?

Note that we can easily talk about smoothness of the underlying function, which we rarely have access to in the real applications. In fact, the general scenario is that we resort to an approximation precisely because we do not have this knowledge.

So far, we have presented problems containing discontinuities as if to mean the underlying function exhibits some discontinuities; e.g., a step function and its variations. However, the PL method is more useful beyond problems that

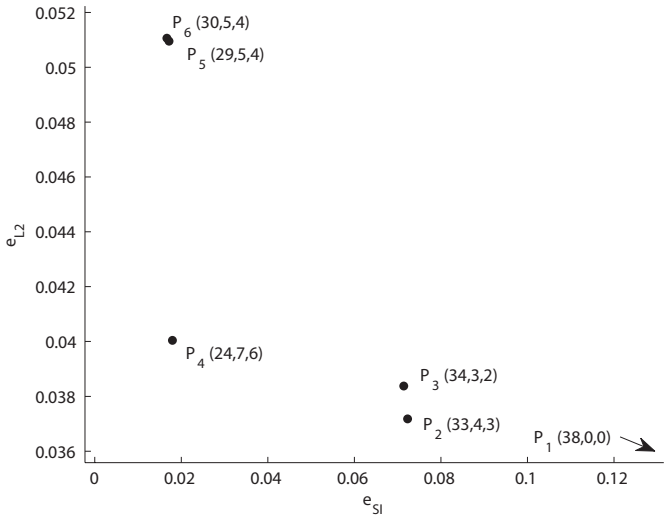


FIG. 6: The Pareto front of PL solutions of the square wave problem with $N = 40$. Note that the exact plot of P_1 is not shown ($e_{\mathrm{SI}} = 1.54$ and $e_{L2} = 0.00259$).

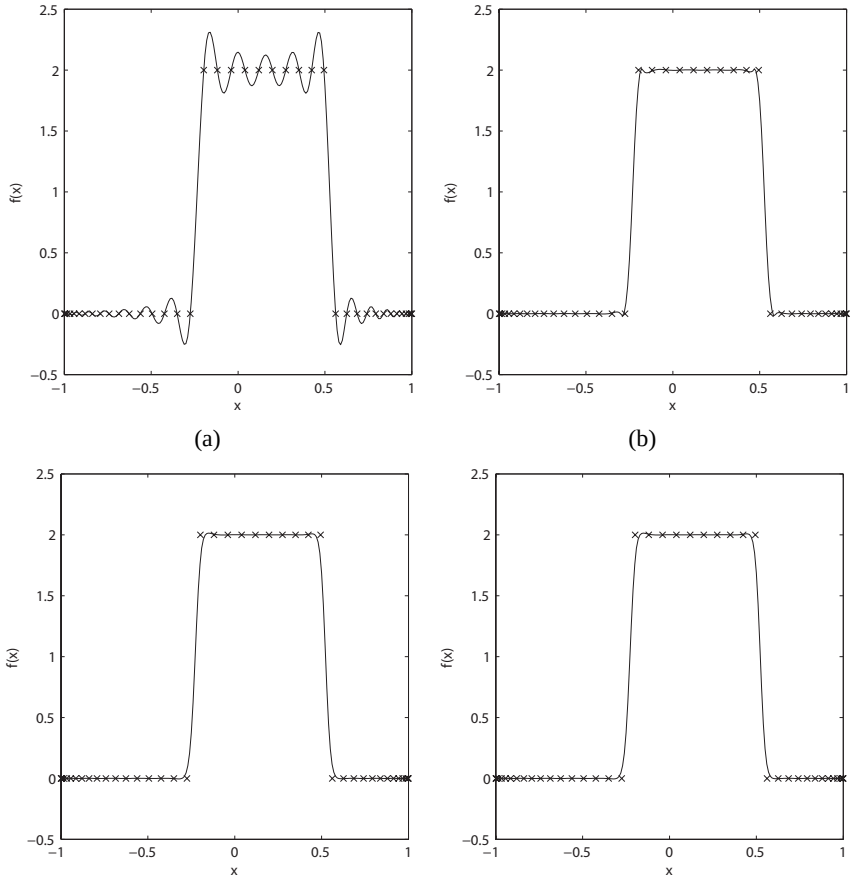


FIG. 7: Some of the PL solutions in the Pareto front for the square wave problem with $N = 40$. (a) P_1 (most accurate), (b) P_2, (c) P_4, and (d) P_6 (smoothest).

contain discontinuities in this strict sense. In the usual absence of knowledge of the underlying function, a steep gradient can be perceived as a discontinuity if the data are coarse enough. In such a case and when the given data are scarce, the present approach proves to be just as useful.

In this section, we explore applications of the PL method on continuous functions with steep gradients. Loosely speaking, for these functions, low-resolution data look discontinuous while high-resolution data look smooth. With the automatic parameter selection in Section 2.1, we numerically show that the PL method degenerates to stochastic collocation when sufficient resolution is achieved.

2.2.1 Smoothness of Discrete Data

One possible way to rigorously define the smoothness of discrete data is through the smoothness of its standard (highest-order) polynomial approximation with respect to the data. More precisely, define the data roughness as the following:

$$\mathrm{DR}(N) = e_{\mathrm{SI}}([M, K, L] = [N, 0, 0]), \tag{17}$$

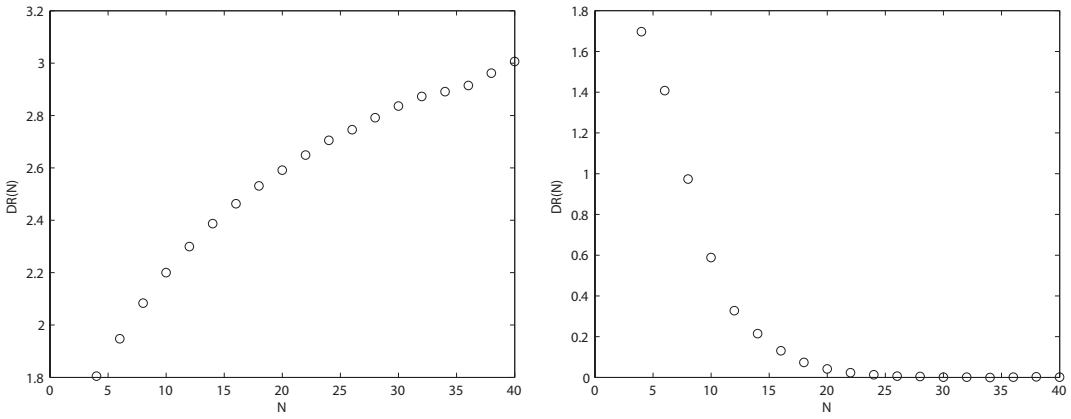
where N is the number of given (LGL) data points and it is implicitly understood what underlying function is being considered. Figure 8(a), shows the data roughness as a function of given data points N for the underlying function $f(x) = \mathrm{sign}(x)$.

Note that f contains a true discontinuity and the data roughness value increases with N. This makes sense because, as more data are revealed, it becomes clearer that the underlying function contains a discontinuity. On the other hand, for a smooth underlying function, the data roughness should decreases with N. Consider a smooth underlying function $f(x) = \tanh(5x)$. The data roughness of this function is shown in Fig. 8(b). Note that the roughness approaches zero as N increases. This makes sense intuitively because, as more data points are included, the data look smoother and smoother.

In the next section, we consider a smooth underlying function and observe the behavior of the PL solutions as N increases.

2.2.2 PL Behavior for Smooth Underlying Functions

Consider a testing function $f(x) = \tanh(x/\delta)$, where δ is a tunable parameter dictating the smoothness of the data sampled from this underlying function. The data roughness as defined in Eq. (17) for this testing function only depends on δ and the number of data points, N.



(a) (b)

FIG. 8: Smoothness of data as a function of the number of data points for (a) $f(x) = \mathrm{sign}(x)$ and (b) $f(x) = \tanh(5x)$. Only the even numbers of data points are shown.

Table 1 shows the data roughness (DR) and the order of the denominator from the PL solutions obtained from the APS algorithm with the lowest e_{L2} (most accurate and least smooth member of the Pareto front) for $\delta = 0.2, 0.3, 0.4$, and N ranging from 4 to 30. Only even N results are shown. The odd N results have similar trends with different starting points and values due to the middle data point being in the region of the steepest gradient.

From Table 1, we observe that when the data are sufficiently smooth, the SC solution ($L = 0$) becomes the most accurate solution. In other words, for smooth underlying functions the PL method (with APS) degenerates to SC when sufficient resolution is achieved. A non-constant denominator does not increase the accuracy, and using all the available data to obtain the numerator part should give the most accurate representation.

More surprisingly, we observe that the order L of the most accurate PL solution increases before the data are smooth enough to yield the SC solution, instead of L decreasing and eventually becoming zero. There is a critical N_{crit} for each smooth underlying function. For $N > N_{\text{crit}}$, the most accurate solution is the SC solution as seen above. For $N < N_{\text{crit}}$, there is a trade-off between increasing M or L for a fixed N, since

$$c(L) + c(M) < c(M + K) \leq c(N), \tag{18}$$

where the first inequality was presented in Section 1.2 and the second is from $M + K \leq N$.

For a fixed $N < N_{\text{crit}}$ and fixed M, we found that increasing L tends to improve the accuracy of the approximated solution. The same is true when increasing M, for a fixed $N < N_{\text{crit}}$ and fixed L. However, it is not clear why both L and M increase as N increases, instead of one parameter dominating the other (one increasing while the other is decreasing). It is difficult to provide a formal explanation of this behavior, especially because it is a result of the quality metrics introduced before and the strategy used to select a set of parameters among all the possible approximant surfaces. However, this behavior will be the subject of further studies.

3. THE PL METHOD IN INVERSION PROBLEMS

This section discusses how PL can be used in Bayesian inference problems. We start by briefly describing the inference problem, then pointing out how PL can be used to accelerate the traditional approach and, finally, concluding with some simple examples to illustrate the methodology.

TABLE 1: Data roughness (DR) and the denominator order (L) of the suggested PL solutions from the APS for the testing function $f(x) = \tanh(x/\delta)$ for various δ and N (the number of data points)

	$\delta = 0.2$		$\delta = 0.3$		$\delta = 0.4$	
N	DR	L	DR	L	DR	L
4	1.698e+0	1	1.361e+0	1	9.532e-1	1
6	1.407e+0	1	7.242e-1	2	2.742e-1	2
8	9.744e-1	2	2.852e-1	2	1.045e-1	2
10	5.882e-1	4	1.474e-1	4	2.627e-2	4
12	3.281e-1	4	6.224e-2	4	7.192e-3	4
14	2.141e-1	6	2.508e-2	6	2.414e-3	0
16	1.311e-1	6	8.718e-3	6	6.083e-4	0
18	7.265e-2	8	3.359e-3	0	2.535e-4	0
20	4.124e-2	8	1.069e-3	0	8.143e-5	0
22	2.352e-2	8	3.840e-4	0	2.603e-5	0
24	1.257e-2	9	1.656e-4	0	8.291e-6	0
26	6.967e-3	0	6.731e-5	0	2.596e-6	0
28	3.665e-3	0	2.839e-5	0	1.143e-6	0
30	1.932e-3	0	1.059e-5	0	8.266e-7	0

3.1 Bayesian Inversion Methodology

In Bayesian inference problems, we are given some observables d and are asked to obtain the unknown input parameters z of the forward model, $G(z)$, that would likely generate those observables. The problem is complicated by the fact that the observables usually are polluted with measurement noise, e:

$$d = d_{\text{true}} + e = G(z) + e, \tag{19}$$

where d_{true} is the solution of the forward model.

Following the Bayesian framework, we have

$$p(z|d) = \frac{p(d|z)p(z)}{\int p(d|z)p(z)dz}, \tag{20}$$

where the shorthand notation $p(z)$ represents the probability density function of the random variable Z at z [subscript Z is omitted as it is clear whose probability density $p(\cdot)$ represents]. Likewise, $p(d|z)$ and $p(z|d)$ are the conditional probability of their corresponding variables. Since the denominator in Eq. (20) is simply a normalization, we can re-express the relationship in terms of proportionality as

$$p(z|d) \propto p(d|z)p(z). \tag{21}$$

We call $p(z)$ the prior probability density and it represents our knowledge of the distribution of Z before we incorporate the data, d. The density of Z conditioned on the data, $p(z|d)$, is called the posterior probability density. Here, $p(d|z)$ is called the likelihood function and, following the independence assumption of the measurement noise e, can be expressed as

$$p(d|z) \triangleq L(z) = \prod_{i=1}^{n_d} p_{e_i}[d_i - G_i(z)], \tag{22}$$

where n_d is the dimension of d. We use notation $L(z)$ for the likelihood function for compactness and to emphasize that it is a function of z.

In this work, we will assume that the measurement noise is described by independent, Gaussian variables with zero mean and σ_e^2 variance. With this assumption, the expression for the likelihood function becomes

$$L(z) = \prod_{i=1}^{n_d} \exp\left(-\frac{[d_i - G_i(z)]^2}{\sigma_e^2}\right) = \exp\left(-\frac{\|d - G(z)\|^2}{\sigma_e^2}\right), \tag{23}$$

where $\|\cdot\|$ is the L_2-norm.

3.2 Using PL in Inversion Problem

The Bayesian framework poses the (inverse) solution as a posterior probability distribution over the input parameters. Although the concept is straightforward, it can be difficult in practice, mainly because the posterior space cannot easily be explored, especially in high-dimensional problems.

To alleviate this problem, several approaches have been proposed, for example, based on sampling [19]; one of the most successful algorithms is based on the Markov chain Monte Carlo (MCMC) method [20, 21]. These approaches require repeated runs of the forward model, and thus when the model is computationally expensive, the method becomes prohibitive.

Many recent works focus instead on the introduction of surrogates for the forward model. In [22], the authors used the stochastic Galerkin method to propagate prior uncertainty through the forward model, thus yielding an approximated forward solution from which the inverse solution can be obtained. In [23], stochastic collocation was used as a

forward model surrogate for posterior evaluation. Similarly, in our work, we employ PL as a surrogate for the forward model:

$$\tilde{G}_N(Z) = \frac{P(Z)}{Q(Z)} = \frac{\sum_{i=0}^{c(M)-1} \hat{p}_i \Psi_i(Z)}{\sum_{i=0}^{c(L)-1} \hat{q}_i \Psi_i(Z)} \approx G(Z). \tag{24}$$

The PL method is better suited for problems with discontinuities and, as seen in Section 2.2, degenerates to standard stochastic collocation when dealing with smooth problems.

3.2.1 Example 1: Step Function

Consider a simple discontinuous forward model

$$G(Z) = \begin{cases} 0 & \text{if } Z \in [-1, 0], \\ 1 & \text{if } Z \in (0, 1], \end{cases} \tag{25}$$

and use one single observation $d = G(z_{\text{true}}) + e$ to define a posterior density $p(z|d)$. The noise e is assumed Gaussian with zero mean and standard deviation of 0.1. The prior distribution on Z is uniform on the entire domain $[-1, 1]$. The original input $z_{\text{true}} = 0.5$, and thus we expect most posterior probability to lie in the right half of the domain.

The SC and PL methods are used to construct surrogates of the forward model, $\tilde{G}_N(Z)$, with $N = 10$ for both methods. Figure 9 shows the resulting posterior densities from the two methods. The SC solution exhibits the oscillatory characteristic of the Gibbs phenomenon as expected, given the discontinuity in the exact forward model. This is undesirable as it suggests variation in probability where none exists. On the other hand, the PL solution is quite uniform across the right half of the domain.

Note that both methods predict low probability in the region $[0, 0.1]$. This is clearly due to low data resolution ($N = 10$).

3.2.2 Example 2: Diffusion Problem

The second example of using PL for the inversion problem involves the following diffusion problem:

$$\frac{\partial u}{\partial t} - \alpha \frac{\partial^2 u}{\partial x^2} = 0$$

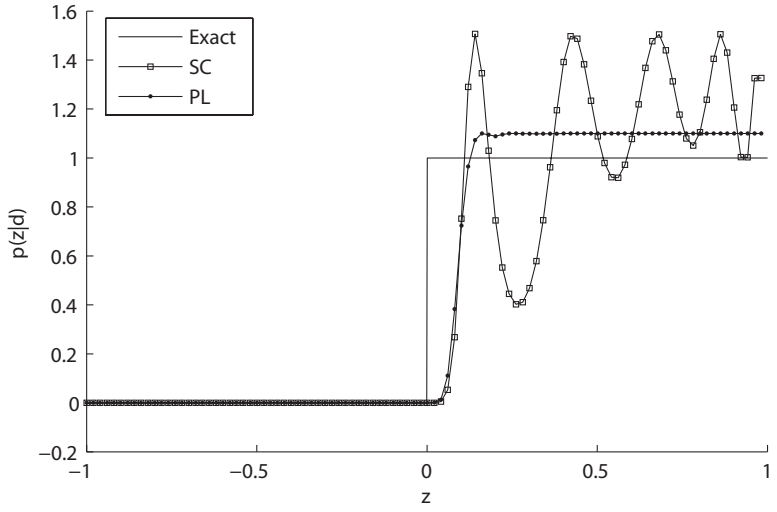


FIG. 9: Exact and approximated posterior density for a step-function $G(Z)$, using SC and PL with $N = 10$. Solid line, exact; dotted line, SC; dashed line, PL.

$$u(0,t) = c(z)$$
$$\frac{\partial u}{\partial x}(1,t) = 0$$
$$u(x,0) = \sin(\pi x) \tag{26}$$

where $\alpha = 1$ and we are interested in the solution at time $t = T = 2$. Here, $c(z)$ is a simple discontinuous function defined as

$$c(z) = \begin{cases} z - 2 & \text{if } z < 0, \\ z & \text{otherwise.} \end{cases} \tag{27}$$

Our forward model is $G(Z) = u(x, T; Z) + e$. We want to infer the unknown input parameter z given a single observation $d(x)$ at $z_{\text{true}} = -0.8$. The prior distribution on Z is uniform in the entire domain $[-1, 1]$. The noise e is assumed to be Gaussian with zero mean and variance of 0.5.

Figure 10 shows the exact solution surface of this problem. The two horizontal axes define the physical and uncertain variables x and z, respectively, and the vertical axis defines the solution $u(x, T; z)$. There is a single sharp ridge along the parameter space at $z = 0$ as a direct result of the discontinuous $c(z)$. For each z, the solution is smooth in the physical coordinate x.

Figures 11(a)–11(f) show the solution surface and posterior density from the SC surrogate with $N = 11, 21$, and 31. From the surface plots, we can clearly observe the Gibbs' oscillation contaminating the solutions away from the ridge, including the region around $z = z_{\text{true}} = -0.8$. This has a significant impact on computation of the posterior density. When $N = 11$ [Figures 11(a) and 11(b)], we observe three separate peaks corresponding to the most likely value of parameter z of the data. By design, we know that the true value of z is at the middle peak. The two side peaks are spurious solutions caused by the discontinuity. As we increase N to 21 and 31, we are left with only one globally maximal peak. However, numerous locally maximal peaks are still undesirable as they are purely artifacts of the SC method and have no direct connection to the given data.

Figures 12(a)–11(f) show the PL solutions for this problem. The data are given at the same locations as those in the SC method above. Note that for both SC and PL the peak gets higher and steeper as we increase N. This shows our increasing confidence in predicting the unknown input parameter as we have more forward runs. Note also that the PL method yields a smooth solution surface for all cases without Gibbs' oscillation. For the same data, the PL

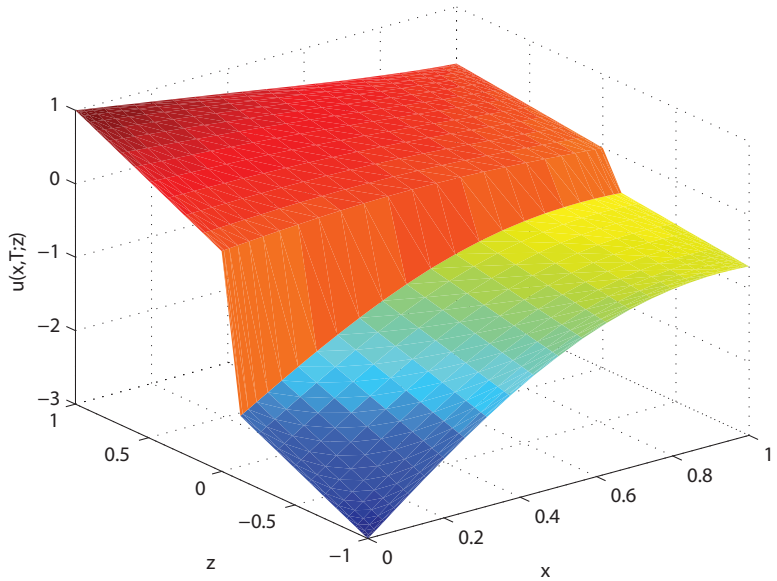


FIG. 10: Exact solution surface of the diffusion problem.

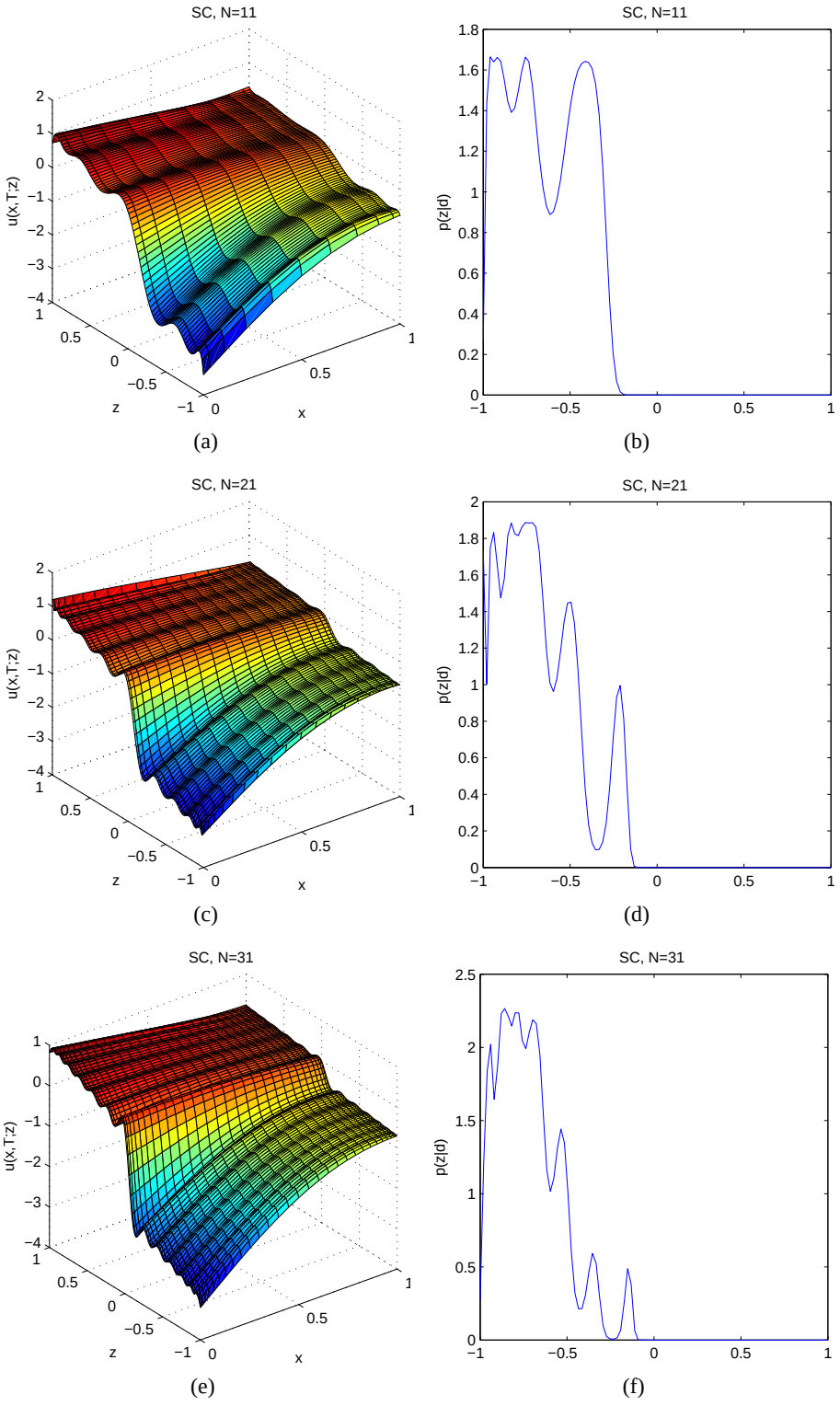


FIG. 11: (a), (c), and (e) Solution surface and (b), (d), and (f) posterior density from SC method with (a) and (b) $N = 11$, (c) and (d) $N = 21$, and (e) and (f) $N = 31$.

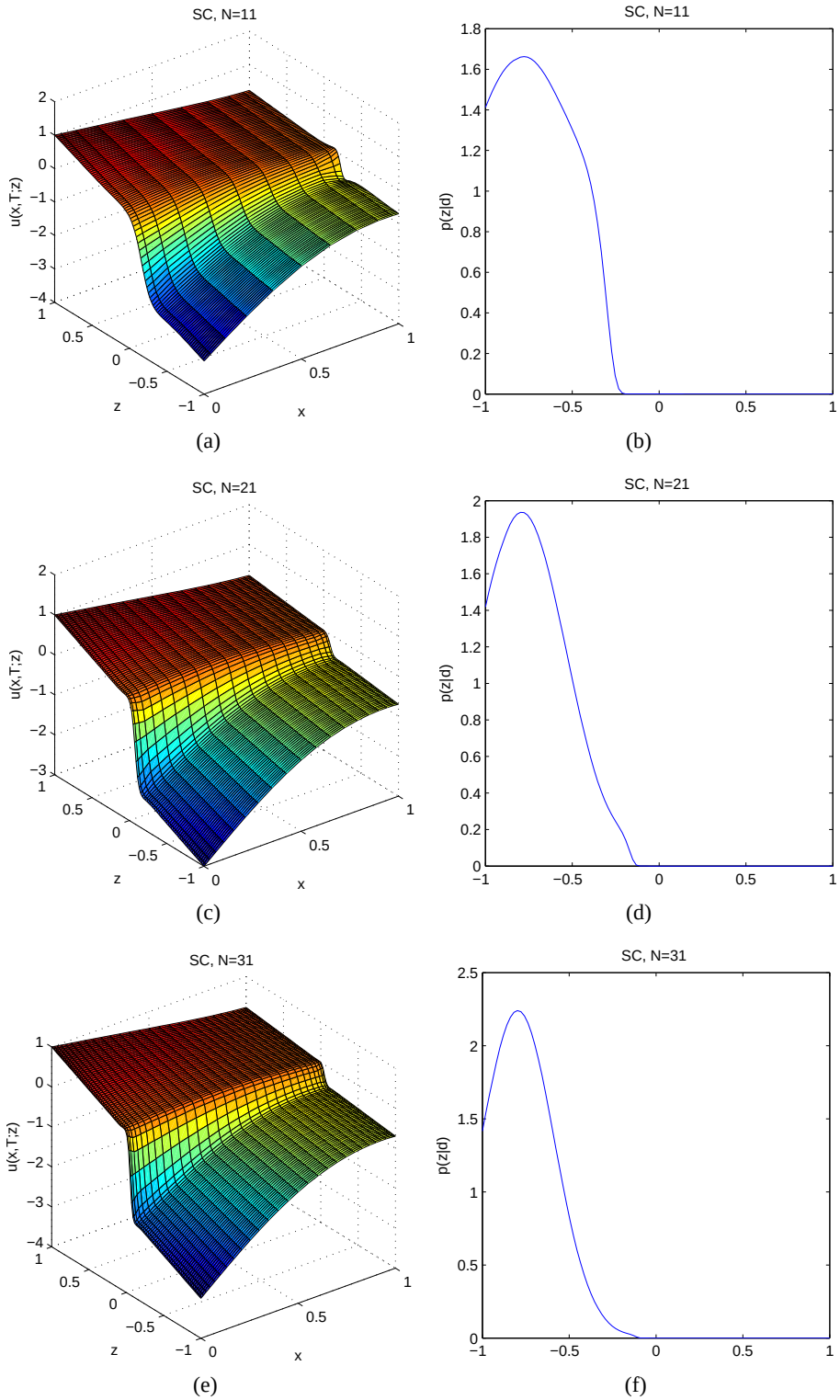


FIG. 12: (a), (c), and (e) Solution surface and (b), (d), and (f) posterior density from SC method with (a) and (b) $N = 11$, (c) and (d) $N = 21$, and (e) and (f) $N = 31$.

method has higher accuracy than the SC method. The PL solution contains one consistent maximal peak around $z_{\rm true}$, while the SC solution contains multiple local maximal peaks.

For a quantitative measure, consider $P_{\rm fail} = P(|z - z_{\rm true}| > e_{\max})$, the probability that the predicted z is too far away from $z_{\rm true}$. In this case, we arbitrarily choose the maximum acceptable difference $e_{\max}$ to be 0.3. Table 2 shows $P_{\rm fail}$ from both SC and PL methods for $N = 11$, 21, and 31. For each N, the PL method gives consistently lower $P_{\rm fail}$, indicating higher accuracy in inferring input parameter z.

TABLE 2: $P_{\rm fail}$ from SC and PL methods with $N = 11$, 21, and 31

N	SC	PL
11	0.337	0.218
21	0.210	0.162
31	0.120	0.099

4. CONCLUSIONS

A novel approach—the Padé-Legendre method—for assessing uncertainty in problems characterized by strong nonlinear and possibly discontinuous system responses has been presented. The method uses a ratio of polynomials to approximate discrete data and can be interpreted as an extension of the popular stochastic collocation approach in which a polynomial interpolant is constructed. In the case, where the lack of smoothness in the data is due to limited resolution, we demonstrated how the PL formulation reverts to a pure polynomial representation as apparent data roughness decreases under increasing data resolution. In addition, we introduced an algorithm to extract a sequence of candidate PL reconstructions that balance the interpolation error and the smoothness of the reconstruction. Finally, we have investigated the use of the PL approach in Bayesian inference, demonstrating that in the presence of discontinuous input parameters, the posterior distributions obtained using polynomial representations can be significantly improved by reducing Gibbs oscillations in the emulators.

ACKNOWLEDGMENTS

The authors wish to thank Dr. Alireza Doostan and Dr. Habib Najm for useful discussions regarding the Padé-Legendre approximation. This material is based upon work supported by the Department of Energy (National Nuclear Security Administration) under Award Number NA28614.

REFERENCES

1. Ghanem, R. and Spanos, P., *Stochastic Finite Elements: A Spectral Approach*, Springer, New York, 1991.
2. Xiu, D. and Karniadakis, G. E., The Wiener-Askey polynomial chaos for stochastic differential equations, *SIAM J. Sci. Comput.*, 24:619–644, 2003.
3. Doostan, A. and Iaccarino, G., A least-squares approximation of partial differential equations with high dimensional random inputs, *J. Computat. Phys.*, 228(12):4332–4345, 2009.
4. Chantrasmi, T., Doostan, A., and Iaccarino, G., Padé-Legendre approximants for uncertainty analysis with discontinuous response surfaces, *J. Computat. Phys.*, 228:7159–7180, 2009.
5. Wang, Q., Moin, P., and Iaccarino, G., A rational interpolation scheme with super-polynomial rate of convergence, *SIAM J. Numer. Anal.*, 47(6):4073–4097, 2010.
6. Xiu, D., Lucor, D., Su, C.-H., and Karniadakis, G., Stochastic modeling of flow-structure interactions using generalized polynomial chaos, *J. Fluids Eng.*, 124:51–59, 2002.
7. Maitre, O. P. L., Najm, H. N., Ghanem, R., and Knio, O. M., Multi-resolution analysis of Wiener-type uncertainty propagation schemes, *J. Computat. Phys.*, 197:502–531, 2004.

8. Wan, X. and Karniadakis, G. E., An adaptive multi-element generalized polynomial chaos method for stochastic differential equations, *J. Computat. Phys.*, 209:617–642, 2005.
9. Constantine, P., Gleich, D., and Iaccarino, G., Spectral methods for parameterized matrix equations, *SIAM. J. Matrix Anal. Appl.*, 31:2681–2699, 2010.
10. Mathelin, L. and Hussaini, M. Y., A stochastic collocation algorithm for uncertainty analysis, *NASA/CR-2003-212153*, 2003.
11. Xiu, D. and Hesthaven, J. S., High order collocation methods for the differential equations with random inputs, *SIAM J. Sci. Comput.*, 27:1118–1139, 2005.
12. Nobile, F., Tempone, R., and Webster, C. G., An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data, *SIAM J. Numer. Anal.*, 46:2309–2345, 2008.
13. Babuka, I., Nobile, F., and Tempone, R., A stochastic collocation method for elliptic partial differential equations with random input data, *SIAM J. Numer. Anal.*, 45 (3):1005–1034, 2007.
14. Salas, M. D., Abarbanel, S., and Gottlieb, D., Multiple steady states for characteristic initial value problems, *Appl. Numer. Math.*, 2:193–210, 1986.
15. Lin, G. and Tartakovsky, A. M., An efficient, high-order multi-element probabilistic collocation method on sparse grids for three-dimensional flow in random porous media, *Proc. of American Geophysical Union, Fall Meeting*, Abstract Number: H23B-1318, 2007.
16. Lin, G., Su, C.-H., and Karniadakis, G. E., Stochastic modeling of random roughness in shock scattering problems: Theory and simulations, *Comput. Methods Appl. Mech. Eng.*, 197:3420–3434, 2008.
17. III, D. B. and Trefethen, L. N., *Numerical Linear Algebra*, SIAM, Philadelphia, PA, 1997.
18. Xiu, D., Efficient collocational approach for parametric uncertainty analysis, *Communi. Computat. Phys.*, 2:293–309, 2007.
19. Gilks, W. R., Richardson, S., and Spiegelhalter, D. J., *Markov Chain Monte Carlo in Practice*, Chapman & Hall/CRC, Boca Raton, FL, 1996.
20. Christen, J. A. and Fox, C., MCMC using an approximation, *J. Computat. Graph. Stat.*, 14(4):795–810, 2005.
21. Higdon, D., Lee, H., and Holloman, C., Markov chain Monte Carlo-based approaches for inference in computationally intensive inverse problems, *Bayesian Stat.*, 7:181–197, 2003.
22. Marzouk, Y. M., Najm, H. N., and Rahn, L. A., Stochastic spectral methods for efficient Bayesian solution of inverse problems, *J. Computat. Phys.*, 224(2):560–586, 2007.
23. Marzouk, Y. M. and Xiu, D., A stochastic collocation approach to Bayesian inference in inverse problems, *Commun. Computat. Phys.*, 6(4):826–847, 2009.