Approximation and error estimation in high dimensional space for stochastic collocation methods on arbitrary sparse samples

Richard K. Archibald[1], Ralf Deiterding[1], Cory Hauck[1], John Jakeman[2], and Dongbin Xiu[3]

[1]Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831
[2]Sandia National Laboratories, Albuquerque, NM 87185
[3]Department of Mathematics, Purdue University, 150 N University St, West Lafayette, IN 47907-2067

E-mail: archibaldrk@ornl.gov

Abstract. We have develop a fast method that can capture piecewise smooth functions in high dimensions with high order and low computational cost. This method can be used for both approximation and error estimation of stochastic simulations where the computations can either be guided or come from a legacy database.

1 Introduction

We have developed a method for generating function approximations with high-order accuracy from arbitrary point-wise values. The method is a variation of stochastic collocation. In the context of uncertainty quantification, each function value represents the quantity of interest for a deterministic run of a complex scientific application and a specified input value. Allowing for function sampling on arbitrary points means, among other things, that the method can be used on legacy databases of application simulations. The method also provides high-order error estimation and edge capturing, can be used to lead ensemble adaptation, and provides the necessary tools to gauge the reliability of information extracted from simulation ensembles. Finally, the *curse of dimensionality* that is problematic for stochastic collocation in high dimensions is mitigated through the use of fast algorithms that scale with $\mathcal{O}\Big(N\ln(N)\Big)$, where N is the number of samples.

2 General Formulation for Interpolation and Edge Detection

Let us first introduce the following notations which will be used throughout this paper. We denote the point in $x \in \mathbb{R}^d$ as $x = (x_1, \cdots, x_d)$. For any finite set of points $\mathcal{S}$ in $\mathbb{R}^d$ we use the notation $K_{\mathcal{S}}$ for the convex hull of the set $\mathcal{S}$. We denote by $\mathbb{N} := \{1, 2, \cdots, \}$ the set of natural numbers and $\mathbb{Z}_+ := \{0, 1, 2, \cdots, \}$ the set of non-negative integers. For any $\alpha \in \{(\alpha_1, \ldots, \alpha_d) : \alpha_1, \ldots, \alpha_d \in \mathbb{Z}_+\} := \mathbb{Z}_+^d$, we set $|\alpha| := \sum_{k=1}^d \alpha_k$, $|\alpha|_{10} := \sum_{k=1}^d 10^{k-1}\alpha_k$, and $\alpha! := \alpha_1! \cdots \alpha_d!$. Throughout α is a multivariate non-negative integer that will change dimension based upon the dimension under discussion. If this multivariate non-negative integer is used as a subscript to a point, x_α, this is used as index to a single point in $\mathbb{R}^d$. However, if α is a superscript, then $x^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_d^{\alpha_d}$.

For any $m \in \mathbb{Z}_+$, Π_m^d denotes the space of all polynomials of degree $\leq m$ in $d \in \mathbb{N}$ variables where the dimension of Π_m^d is denoted by

$$m_d := \binom{m+d}{d}. \tag{1}$$

For $\alpha, \beta \in \mathbb{Z}_+^d$

$$\binom{\alpha}{\beta} = \binom{\alpha_1}{\beta_1} \cdots \binom{\alpha_d}{\beta_d}. \tag{2}$$

Finally, we recall the kronker delta function

$$\delta_{i,j} = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{if } i \neq j. \end{cases} \tag{3}$$

We present here the formulation and key properties of the block multivariate Lagrange interpolation method studied in [7]. The notation introduced in [7] is modified to conform to the new methods presented in this paper. A *block* of points is denoted as

$$\mathbf{x}^m = \{x_\alpha : |\alpha| = m\}, \quad \text{for } m = 0, 1, 2, \ldots. \tag{4}$$

A set of interpolation points, irregularly distributed points in a bounded domain $\Omega \subset \mathbb{R}^d$, is given in block notation as

$$\mathcal{S}_{m_d} := \{\mathbf{x}^0, \mathbf{x}^1, \ldots, \mathbf{x}^m\}. \tag{5}$$

If $\mathcal{S}_{m_d}$ is poised, then there exist polynomials $q_\alpha \in \Pi_{|\alpha|}^d$ for $|\alpha| \leq m$, such that

$$q_\alpha(x_\beta) = \delta_{\alpha,\beta}, \quad |\alpha|, |\beta| \leq m \tag{6}$$

If these polynomials exist, they are called the *Lagrange fundamental* polynomials and interpolation is given for a function f sampled on points in $\mathcal{S}_{m_d}$ by,

$$f_{\mathcal{S}_{m_d}}(x) = \sum_{x_\alpha \in \mathcal{S}_{m_d}} f(x_\alpha) q_\alpha(x) \tag{7}$$

The *Lagrange fundamental* polynomials are given explicitly as

$$q_\alpha(x_\beta) = \frac{\tau_\alpha^m(X|x)}{\tau^m(X)} \tag{8}$$

where $\tau^m(X)$ is the determinant of the Vandermonde matrix

$$X = [x_\alpha^\beta], \tag{9}$$

and $\tau_\alpha^m(X|x)$ is the determinant of the vandermonde matrix where x_α is replaced by x. The interpolation problem given in (7) is called poised when Vandermonde matrix (9) has a nonvanishing determinant.

The Newton formula for the Lagrange interpolation polynomials use the *Newton fundamental* polynomials, $p_\alpha \in \Pi_{|\alpha|}^d$, and are defined by

$$p_\alpha(x_\beta) = \begin{cases} 0, & \text{if } |\beta| < |\alpha|, \\ \delta_{\alpha,\beta}, & \text{if } |\beta| = |\alpha|. \end{cases} \tag{10}$$

The Newton fundamental polynomials can be given for each $|\alpha| = n \leq m$ using only a reduced vandermonde matrix from the set of points $\{\mathbf{x}^0, \mathbf{x}^1, \ldots, \mathbf{x}^n\}$ and given explicitly as

$$p_\alpha(x_\beta) = \frac{\tau_\alpha^n(X|x)}{\tau^n(X)} \tag{11}$$

The finite difference Newton formula for interpolation is given by

$$f_{\mathcal{S}_{m_d}}(x) = \sum_{x_\alpha \in \mathcal{S}_{m_d}} \lambda_{|\alpha|}[\mathbf{x}^0, \mathbf{x}^1, \ldots, \mathbf{x}^m] f \cdot p_\alpha(x) \tag{12}$$

with remainder formula

$$f(x) - f_{\mathcal{S}_{m_d}}(x) = \lambda_{m+1}[\mathbf{x}^0, \mathbf{x}^1, \ldots, \mathbf{x}^m, x] f, \tag{13}$$

where the block finite difference is defined recursively as,

$$\begin{aligned} \lambda_0[x] f &= f(x) \\ \lambda_{m+1}[\mathbf{x}^0, \mathbf{x}^1, \ldots, \mathbf{x}^m, x] f &= \lambda_m[\mathbf{x}^0, \mathbf{x}^1, \ldots, \mathbf{x}^{m-1}, x] f \\ &\quad - \sum_{|\alpha|=m} \lambda_{|\alpha|}[\mathbf{x}^0, \mathbf{x}^1, \ldots, \mathbf{x}^{m-1}, x_\alpha] f \cdot p_\alpha(x). \end{aligned} \tag{14}$$

An edge detection method on the set of irregularly distributed points in a bounded domain Ω in $\mathbb{R}^d$ was introduced in [2]. Let $\mathcal{S}$ be a set of discrete points in Ω and f be a piecewise smooth function known only on $\mathcal{S}$. In order to identify the jump discontinuities of f, we construct a function $L_{\mathcal{S}_{m_d}} f$, $m \in \mathbb{N}$ and $\mathcal{S}_{m_d}$ (5), which can be characterized by the asymptotical convergence property,

$$L_{\mathcal{S}_{m_d}} f(x) \longrightarrow 0,$$

for any x away from discontinuities, with the convergence rate depending in part on the given positive integer m which influences the size of the local stencle, $\mathcal{S}_{m_d}$. The choice of m is user dependent but higher number m provides a faster rateof convergence in smooth regions of f.

The edge detection method presented here is based on a local polynomial annihilation property. The general form of $L_{\mathcal{S}_{m_d}} f$ is given by the following two step method. In the first step, for any $x \in \Omega$, we choose a local set, $\mathcal{S}_{m_d}$ (5), of m_d (1) points around x.

In order to annihilate polynomials up to degree $m-1$, we solve a linear system for the coefficients $c_\alpha(x)$ given by

$$\sum_{x_\alpha \in \mathcal{S}_{m_d}} c_\alpha(x) p_\beta(x_\alpha) = \sum_{|\alpha|=m} p_\beta^{(\alpha)}(x), \quad \alpha \in \mathbb{Z}_+^d, \tag{15}$$

where set of p_β form a basis for Π_m^d. Note that the solution (15) exists and is unique. Our edge detector $L_{\mathcal{S}_{m_d}} f$ is defined using the solution of (15) as

$$L_{\mathcal{S}_{m_d}} f(x) = \frac{1}{N_{m,d}(x)} \sum_{x_\alpha \in \mathcal{S}_{m_d}} c_\alpha(x) f(x_\alpha). \tag{16}$$

Here $N_{m,d}(x)$ is a suitable normalization factor depending on m, the dimension d, and the local set $\mathcal{S}_{m_d}$ (5).

3 Connecting Adaptation, Interpolation, and Error Estimation

The previous section demonstrates how interpolation can be done explicitly for poised sets $\mathcal{S}_{m_d}$. There exist fast algorithms for interpolation evaluation of these explicit formulations [4]. Next we present a theorem that provides an explicit formulation of global approximation of the interpolation error.

Theorem 3.1. *Given a poised set $\mathcal{S}_{m_d}$ with interpolation formulation give in (12) the interpolation error is given explicitly as*

$$\begin{aligned} f(x_\alpha) - f_{\mathcal{S}_{m_d}\setminus x_\alpha}(x_\alpha) &= \frac{N_{m,d}(x_\alpha)}{c_\alpha} L_{\mathcal{S}_{m_d}} f(x_\alpha), \\ &= \frac{1}{c_\alpha} \sum_{x_\alpha \in \mathcal{S}_{m_d}} c_\alpha(x) f(x_\alpha) \end{aligned} \tag{17}$$

where $L_{\mathcal{S}_{m_d}} f$ is the edge detector given in (16)

The proof of Theorem 3.1 is given in [1]. This theorem demonstrates how the edge detector given in (16) can be modified to give the exact error for every $x_\alpha \in \mathcal{S}_{m_d}$ of the function evaluation at x_α as compared to the evaluation of the interpolation formulation give in (12), where the interpolation is based on the set $\mathcal{S}_{m_d} \setminus x_\alpha$. The power of this result is that given a set $\mathcal{S}_{m_d}$, by solving for the edge detection coefficients (15), error estimates can be given nodally for every member of $\mathcal{S}_{m_d}$ providing an estimate around nodally sampled points. Naturally, regions with large error estimates would be prime candidates for further adaptation.

The results of Theorem 3.1 can be extended to sets of arbitrary length [1]. We build the algorithms in this paper around interpolation and error estimation of sets of arbitrary length.

4 Mitigating the Curse of Dimensionality

The *curse of dimensionality* comes about when the complexity of a method grows exponentially with respect to the dimension. This problem is especially relevant for stochastic collocation methods since dimension size grows with the number of random variables in the simulated system. The benefit of the multivariate Lagrange interpolation given in (12) is that for a smooth function, f, sampled on a set of N points $\mathcal{S} \subset \mathbb{R}^d$ a high order reconstruction can be achieved with the largest computational cost coming from Vandermonde matrix determinate which can be calculated in $\mathcal{O}\Big(N \ln(N)\Big)$ [5]. However, suppose that f is a piecewise smooth function. High order

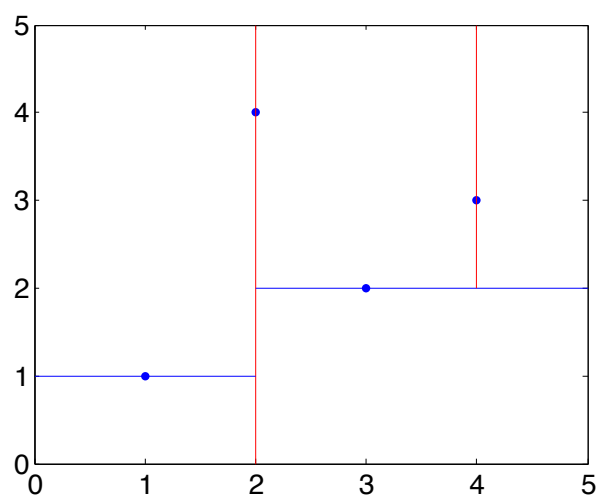

Figure 1: Two dimensional example of a k-d tree. First node at $(2,4)$ performs a binary split of the data along a hyperplane in the first dimension. Second level nodes $(1,1)$ and $(3,2)$ performs a binary split of the data in their branch along a hyperplane in the second dimension. Dimensional binary splits cycle among dimensions until no points remain in new branches.

reconstruction is possible if multivariate Lagrange interpolation is done on smooth sub-regions. The problem becomes how to partition $\mathcal{S}$ into smooth sub-regions.

Suppose we are given a piecewise smooth function, f, sampled on a set of N points $\mathcal{S} \subset \mathbb{R}^d$. A high order reconstruction of f is possible if we use the multivariate Lagrange interpolation given in (12) to reconstruct in smooth sub-regions. The problem becomes how to partition $\mathcal{S}$ into smooth sub-regions.

Given any suitable partition, $\{\mathcal{S}_1, \ldots, \mathcal{S}_M\}$, of $\mathcal{S}$ we can use Theorem 3.1 to provide an error estimate of the multivariate Lagrange interpolation. A suitable partition would require that $K_{\mathcal{S}_i} \bigcap K_{\mathcal{S}_j} = \emptyset$ for all $i \neq j$. Given that the number of points in the ith partition is N_i, the error estimation of that partition can be solved using the *Lagrange fundamental* polynomials (8) at a cost of $\mathcal{O}\Big(N_i\Big)$[1]. Thus, given any partition the error estimation of the entire partition will cost $\mathcal{O}\Big(N\Big)$. Developing a suitable partition requires that convex hull of any two partitioned sets do not overlap. A notion of how points in $\mathcal{S}$ are connected is required to develop a suitable partition. One well studied method to map the connections of points $\mathbb{R}^d$ is by delaunay triangulation, which for poised that do not lie on lower dimensional manifold, would have a computational and memory cost of $\mathcal{O}\Big(N_i^{\lceil \frac{d}{2} \rceil}\Big)$ [6]. This method, like many partitioning techniques, suffer from the *curse of dimensionality*. We use a k-d tree, first introduce in [3], to search for the best suitable partition. The generation of the k-d tree can be done in $\mathcal{O}\Big(N \log(N)\Big)$ and we use the form of the k-d tree to generate fast partitioning and edge representation algorithms. A two dimensional example is given in Figure 1 to explain the basic concepts around the k-d tree algorithm.

5 Fast Algorithms for Adaptation, Interpolation, and Error Estimation

Based on the k-d tree algorithm we give a fast algorithm to generate a suitable partition of a piecewise smooth function, f, on a finite set of sampled points $\mathcal{S} \subset \mathbb{R}^d$.

algorithm 5.1 (Fast suitable partition)**.** *Given a poised set $\mathcal{S} \subset \mathbb{R}^d$ that samples a piecewise smooth function f, initialize the binary partition dimension to one, the partition as $\mathcal{S}$, and set some smoothness criterion, τ_s, for the smoothness estimator (17).*

1. *Estimate smoothness of $\mathcal{S}$. If $\max_{x_\alpha \in \mathcal{S}} \frac{1}{c_\alpha} \sum_{x_\alpha \in \mathcal{S}} c_\alpha(x) f(x_\alpha) \leq \tau_s$ then smoothness requirement is satisfied and exit, else split $\mathcal{S}$ along the binary partition dimension, cycle the binary partition dimension and continue.*

2. *Estimate smoothness of all newly split partitions. If partition is smooth, merge to neighboring partitions that have a combined smooth estimate. If partition is not smooth and has more that one point, split along the binary partition dimension.*

3. *If new partitions have been created, cycle the binary partition dimension and repeat step 2, else exit.*

The computational cost for Algorithm 5.1 will be the cost of estimating the smoothness of a partition times the number of partitions that are estimated. The binary partitioning scheme of this algorithm combined with fast error estimation produces a computational cost for Algorithm 5.1 of $\mathcal{O}\Big(N \ln(N)\Big)$. Finally, a binary partition of a set of points consists of splitting the set at the median point in a list of points that is sorted along the binary partition dimension.

[1] Using these polynomials results in a diagonal system with the only cost coming from finding the derivatives

The execution of Algorithm 5.1 will produce a suitable partition and the error estimate for this partition. The edges in between partitions and domain boundaries are also required to fully characterize the high order approximation of a piecewise smooth function. Below we present an algorithm that $\mathcal{O}\Big(N\ln(N)\Big)$ that uses the suitable partition to characterizes theses edges.

algorithm 5.2 (Fast edge representation)**.** *Given a suitable partition, $\{\mathcal{S}_1,\ldots,\mathcal{S}_M\}$, of $\mathcal{S}$, initialize the binary partition dimension to one. The edges between partitions can be determined by*

1. *If the suitable partition contains only $\mathcal{S}$, return no edge, else continue.*
2. *Perform k-d* tree *algorithm, terminating each branch if the subset branch is contained completely in $\mathcal{S}_i$ for some $i = 1,\ldots,M$.*
3. *Edges are determined by merging branches of the partial k-d* tree *that are in the same partition.*
4. *Margins around these edges can be determined by finding the closest data point in $\mathcal{S}$ normal to each edge and the total volume of all margins is given as V_m.*

The important measure from Algorithm 5.2 is the total volume of all margins V_m. This measure gives a estimate to how well the edges between piecewise smooth regions can be approximated. A suitable stoping criteria for adaptation with respect of edge estimation is that the volume of the margins fall below some given threshold, or $V_m \leq \tau_v$.

To demonstrate these algorithms and approach, consider the following example.

Example 5.1.

$$f_1(x) := \begin{cases} \sin\left(\pi x\right) & 0 \leq \|x\| < \frac{1}{2}, \\ -\sin\left(\pi x\right) & \frac{1}{2} \leq \|x\| \leq 1. \end{cases} \tag{18}$$

Figure 2 displays the progression of the high order approximation of this piecewise smooth function. Given an initial set of random points, Figure 2a displays the suitable partition generated by Algorithm 5.1. Using this partition, Figure 2b displays the edge representation of the suitable partition generated by Algorithm 5.2. For each partition there is an inner and outer edge representation, yielding a marginal area. Adaptation proceeds by adding new points that will reduce the marginal areas and reduce the error estimates in the smooth partitions. The final partition is given in Figure 2c, were the stoping criteria of is satisfied with a total of 24 points. The stoping criteria for adaptation in smooth regions is a L_2 error estimate of less than $\tau_s = 2e^{-2}$ and a marginal area of less than $\tau_v = 5\%$. The final L_2 error estimate was $1.9e^{-2}$, the actual error was $1.5e^{-2}$. High accuracy with low point density is maintained by having two stopping requirements. The first stopping criteria, τ_s, controls the convergence in smooth regions, which can be approximated well with polynomials. The second criteria, τ_v, controls the margin which contains the edges. The margin gives a robust measure of the edge position, and prevents dense sampling around these edges.

Acknowledgments

[1] R. Archibald, R. Deiterding, C. Hauck, J. Jakeman, and D. Xiu. Approximation and error estimation in high dimensional space for stochastic collocation methods on arbitrary sparse samples. *In preparation*, 2012.

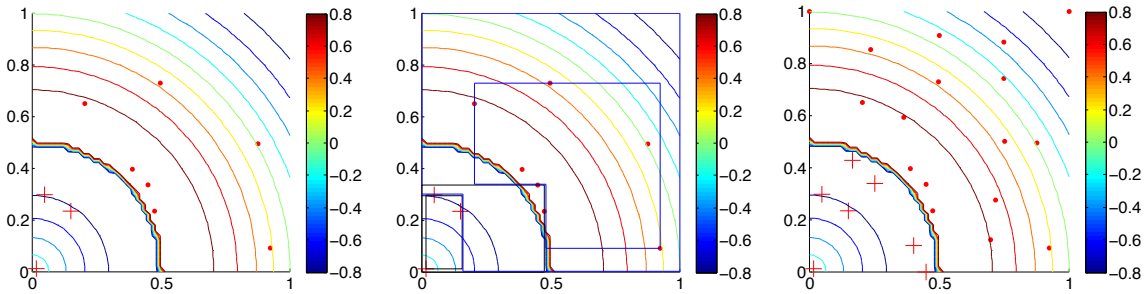

Figure 2: (a) Red points and crosses represent a suitable partition generated by Algorithm 5.1 on randomly sampled points. The contour plot of equation (5.1) is given as a reference. (b) Edge representation of the suitable partition generated by Algorithm 5.2, where the two black contours represent the inner and outer edge of the red cross partition and the two blue contours represent the inner and outer edge of the red point partition. (c) Final partition after adaptation reaches stoping criteria.

[2] R. Archibald, A. Gelb, and J. Yoon. Polynomial fitting for edge detection in irregularly sampled signals and images. *SIAM journal on numerical analysis*, 43(1):259, 2006. 0036-1429.

[3] H. Friedman Jerome, Bentley Jon Louis, and Finkel Raphael Ari. An algorithm for finding best matches in logarithmic expected time. *ACM Trans. Math. Softw.*, 3(3):209–226, 1977. 355745.

[4] Joanna Kapusta and Ryszard Smarzewski. Fast algorithms for multivariate interpolation and evaluation at special points. *Journal of Complexity*, 25(4):332 – 338, 2009.

[5] Lei Li and Tadao Nakamura. Fast parallel algorithms for vandermonde determinants. *International Journal of Computer Mathematics*, 73(4):479–486, 2000. doi: 10.1080/00207160008804911 0020-7160 doi: 10.1080/00207160008804911.

[6] Amenta Nina, Attali Dominique, and Devillers Olivier. Complexity of delaunay triangulation for points on lower-dimensional polyhedra, 2007. Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms.

[7] T SAUER and Y XU. On multivariate lagrange interpolation. *MATH. COMP*, 64:1147–1170, 1994.