Proceedings of the ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference
IDETC/CIE 2016
August 21-24, 2016, Charlotte, North Carolina

DETC2016-60195

A COMPRESSED SENSING APPROACH TO UNCERTAINTY PROPAGATION FOR APPROXIMATELY ADDITIVE FUNCTIONS

Kaiyu Li and Douglas Allaire*
Computational Design Laboratory
Department of Mechanical Engineering
Texas A&M University
College Station, TX 77843
Email: {kaiyuli, dallaire}@tamu.edu

ABSTRACT

Computational models for numerically simulating physical systems are increasingly being used to support decision-making processes in engineering. Processes such as design decisions, policy level analyses, and experimental design settings are often guided by information gained from computational modeling capabilities. To ensure effective application of results obtained through numerical simulation of computational models, uncertainty in model inputs must be propagated to uncertainty in model outputs. For expensive computational models, the many thousands of model evaluations required for traditional Monte Carlo based techniques for uncertainty propagation can be prohibitive. This paper presents a novel methodology for constructing surrogate representations of computational models via compressed sensing. Our approach exploits the approximate additivity inherent in many engineering computational modeling capabilities. We demonstrate our methodology on an analytical function and a cooled gas turbine blade application. The results of these applications reveal substantial computational savings over traditional Monte Carlo simulation with negligible loss of accuracy.

1 INTRODUCTION

Increasingly, computational models of physical systems are being used to support decision-making processes. These processes include design decisions, policy level analyses, and experimental design decisions among other things. Typically, computational models have uncertainty associated with their inputs and parameters, which subsequently propagates through the model, begetting uncertainty in the outputs, or quantities of interest, the model estimates. Therefore, the application of model output information to support decision-making processes requires the effective propagation of uncertainty. The process of propagating uncertainty from model inputs to model outputs could be conducted, for example, via Monte Carlo simulation, but this approach traditionally requires several thousand model evaluations. However, computational models of physical systems of practical use in engineering decision-making processes are often expensive in terms of the time it takes to compute a set of outputs given a set of inputs. Therefore, conducting several thousand model evaluations for something like Monte Carlo simulation, is often computationally intractable.

For the case of computationally expensive models, recourse is often made to surrogate modeling techniques, both through approximations of the underlying physics of the modeling capability, and the underlying uncertainty of the model inputs. We present here, a novel surrogate modeling approach for uncertainty propagation constructed from the powerful concept of compressed sensing. Given a functional basis representation for which the output of a computational model is sparse, we demonstrate that we can recover near-exact approximations of the computational model as a function of its inputs with remarkably few

*Address all correspondence to this author.

function evaluations. The surrogate representation of the model can then be used with any supported probability distribution to provide rapid uncertainty propagation results.

Our approach is specifically catered to the class of what we refer to as *approximately additive* functions. By this, we mean functions that depend primarily on additive terms of subfunctions of low dimension. For example, a function of many inputs may be adequately represented by a sum of subfunctions of each individual input and a few subfunctions of two input combinations (i.e., interaction terms). As discussed in the following section, this approximately additive characteristic is believed to be a feature of a wide-variety of computational models often used for engineering purposes.

This paper proposes a systematic method to apply compressed sensing to approximately additive functions to generate surrogate models for use in uncertainty propagation at a fraction of the cost of traditional Monte Carlo simulation. Our method includes a greedy approach to determining which subfunctions of a given computational model to interrogate for inclusion based on approximate main effect global sensitivity indices. We estimate these indices using the current subfunctions in the surrogate representation. Our method also incorporates a randomized cross validation-based error measure to monitor convergence of the surrogate model estimates to the full computational model estimates. In this work, we assume a polynomial basis representation is sufficient to identify a sparse representation of a given computational model. It is a topic of future work to identify appropriate functional bases for general computational models. We demonstrate our methodology on an analytical test problem, as well as a finite element model of a gas turbine blade with cooling channels. In Section 2, we provide background on the state of the art in uncertainty propagation, the mathematical concept of an additive function, and on compressed sensing. In Section 3, we describe our methodology for constructing a sparse surrogate approximation of a computational model. In Section 4, we demonstrate our approach on an analytical model and the cooled gas turbine blade. Conclusions and topics for future work are then discussed in Section 5.

2 BACKGROUND

Uncertainty propagation, in the context of computational modeling, involves mapping the uncertainty in the inputs to uncertainty on the outputs of the computational model. Without loss of generality, we will deal with computational models that yield a single output, or single quantity of interest. The process of mapping uncertainty from inputs to model output often involves a sample-based approach to gathering input settings that are then either mapped through the full computational model by evaluating the model output at each input setting, or are mapped through an approximation, or surrogate model, of the computational model.

The most basic and most common approach to propagating uncertainty through computational models is via Monte Carlo simulation. For a general computational model, $f(\mathbf{X})$, where $\mathbf{X} = (X_1, X_2, \ldots, X_d)^T$, and the $\mathbf{X}$ is a random vector, Monte Carlo simulation works by sampling a point $\mathbf{x}$, from the distribution of $\mathbf{X}$, and then running the computational model to evaluate $f(\mathbf{x})$. If this process is repeated several times (usually thousands), then the strong law of large numbers guarantees that the empirical distribution of the output evaluations converges almost surely to that of $f(\mathbf{X})$ [1]. That is

$$\hat{F}_n(\mathbf{t}) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_i(\mathbf{t}) \xrightarrow{a.s.} F(\mathbf{t}) \text{ as } n \to \infty, \tag{1}$$

where $\hat{F}_n$ is the empirical distribution of $f(\mathbf{X})$ generated by sampling, F is the cumulative distribution function of $f(\mathbf{X})$, $\mathbf{1}_i(\mathbf{t})$ is the indicator function for the event, $\{x_j^i \leq t_j, \forall j \in \{1,2,\ldots,d\}\}$, where x_j^i is the i-th sample of the j-th input. Given this convergence property, Monte Carlo simulation, with enough input samples, is often considered the gold standard to compare against when developing new algorithms for uncertainty propagation. It has also seen wide use in many engineering and scientific applications [2–8]. However, the convergence rate, which is governed by the central limit theorem, is $O(1/\sqrt{n})$, thus making the method impractical for most expensive computational models for uncertainty propagation [9].

To improve upon the convergence performance of traditional Monte Carlo simulation, different sampling strategies have been studied and applied, such as quasi-Monte Carlo methods, stratified sampling, and Latin hypercube sampling [10]. Quasi-Monte Carlo methods are built off the use of low discrepancy sequences, which are designed to fill the input space as uniformly as possible (i.e., with low discrepancy) [11]. Convergence of these methods is $O((\log n)^d/n)$, and is thus generally superior to traditional Monte Carlo simulation when the number of inputs, d, is not too large. Latin hypercube sampling, or LHS [12] operates by sampling on what is referred to as a Latin hypercube, which requires each sample to be the only sample in each hyperplane it belongs to in the hypercube. Stratified sampling proceeds by placing samples in certain subsets, or strata, in a sample space, and then weighting these points by the probability of being in that particular stratum. Convergence rates of Latin hypercube and stratified sampling methods tend to be superior to Monte Carlo simulation for low input dimension, but degrade as the number of inputs increase.

For expensive computational models, the use of sample-based approaches that use the full model are often computationally prohibitive. A typical strategy in these circumstances is to create a surrogate or metamodel of the full model using a small set of samples from the full model. Often data-fit techniques,

such as response surface modeling or the use of Gaussian processes are employed to construct such models [13–15]. Other alternatives are hierarchical surrogate models [16,17] that consider hierarchies of modeling assumptions or computational grids, and reduced-order models [18] that rely on underlying knowledge of the governing equations of the computational model. The result of developing a surrogate model is usually a much cheaper, in a computational sense, version of the computational model that can be used for various purposes, such as uncertainty propagation. The reduction in runtime, however, usually comes at the expense of some loss in accuracy, or fidelity, in the surrogate model. Another alternative to dealing with an expensive computational model in uncertainty propagation, is the use of a surrogate representation of the uncertainty in the inputs. Some techniques for this include implicit uncertainty propagation, moment matching, the advanced mean value method, and unscented transforms [19–24].

The approach we propose here for propagating uncertainty through a computational model involves some aspects of sample-based methods and the development of a data-fit surrogate model via projection of the data onto a functional basis representation. To ensure scalability of our approach with increasing input dimension, we rely on an assumption of approximate additivity of the underlying full computational model. To introduce this concept, we consider first the high-dimensional model representation (HDMR) of a function, $f(\mathbf{x})$, which is written as [25–27],

$$\begin{aligned} f(x_1,x_2,\ldots,x_d) =& f_0 + \sum_{j=1}^{d} f_j(x_j) + \sum_{j<l}^{d} f_{j,l}(x_j,x_l) + \cdots \\ &+ f_{1,2,\ldots,d}(x_1,x_2,\ldots,x_d) \\ =& \sum_{\mathbf{u}\subseteq\mathscr{D}} f_{\mathbf{u}}(\mathbf{x}_{\mathbf{u}}), \end{aligned} \tag{2}$$

where $\mathscr{D} := \{1,2,\ldots,d\}$ denotes the set of input indices, $\mathbf{u}$ is a multi-index, and individual terms in each summand are referred to as subfunctions. For square-integrable functions, which is the class of functions we are concerning ourselves with in this paper (i.e., those functions with finite variance when the inputs are random variables), we can write the function's variance as

$$\mathbb{V}(f) = \sum_{\substack{\mathbf{u}\subseteq\mathscr{D} \\ \mathbf{u}\neq\emptyset}} \mathbb{V}(f_{\mathbf{u}}), \tag{3}$$

which is a sum of each individual subfunction's variance. In variance-based global sensitivity analysis, sensitivity indices for different combinations of inputs are estimated by normalizing each individual subfunction's variance by the total variance of f [28]. We make use of this concept in our greedy approach to determining which subfunctions to include in our surrogate representation as discussed in Section 3. Given the variance of a function as written above, we can define the effective superposition dimension of the function. Following Ref. [29], the effective superposition dimension is the smallest integer, d_s, such that

$$\sum_{\substack{|\mathbf{u}|\leq d_s \\ \mathbf{u}\neq\emptyset}} \mathbb{V}(f_{\mathbf{u}}) \geq \alpha\mathbb{V}(f), \tag{4}$$

where α is a user defined constant, such as 0.99. If a function has a low effective superposition dimension, then higher order interactions are not important in the construction of the HDMR in Eq. 2. It turns out, that a great deal of functions, particularly in the finance community, but also in engineering, have low effective dimension [7,8,29–31].

For functions with low effective dimension, e.g., $d_s = 2$, then the function is approximately additive in the sense that it consists primarily of subfunctions that are a function of only one input, with a few potentially significant subfunctions of more than one input. We seek to exploit the approximate additivity of many computational models by using the tools of compressed sensing to interrogate the low order subfunctions of a given function, f, and determine if the function is well-represented as approximately additive. If so, then uncertainty propagation can be carried out with a surrogate model constructed by summing the low-order subfunctions found by compressed sensing, which, as shown in Section 4, results in substantial computational savings as compared to traditional Monte Carlo methods with nearly zero loss of accuracy.

Compressed sensing is a recently developed technique for recovering sparse signals that relies on linear dimensionality reduction [32–34]. We provide a brief overview of the rich field here. The general concept, as we employ it here, is that certain signals (for us outputs of computational models) can be approximated well by a sparse representation in a particular functional basis. Thus, the coefficient vector in the functional basis requires only a few nonzero entries. For a given set of basis functions, $\{\psi_k\}_{k=1}^N$, we assume that a signal, f, can be represented as a linear combination

$$f = \Psi\mathbf{c}, \tag{5}$$

where Ψ is an $N\times N$ matrix with columns, ψ_k, and $\mathbf{c}$ is an $N\times 1$ vector of coefficients. If f is sparse (or approximately sparse) in the basis, Ψ, then $\mathbf{c}$ will consist of many values that are effectively zero. The function is called S-sparse in Ψ if there exists a $\mathbf{c}\in\mathbb{R}^N$ with only $S << N$ nonzero entries. Samples of the signal, f, are obtained by another linear operator, Φ, which is an $M\times N$ measurement matrix, where $M < N$. A requirement of

compressed sensing is that Φ and Ψ be as incoherent as possible, meaning as dissimilar as possible [34]. We accomplish this here by setting Φ equal to a random subset of the rows of the identity matrix. Then the sampled signal is

$$\mathbf{b} = \Phi f, \tag{6}$$

which in our context, is just M (rather than N, or even n in the context of Monte Carlo simulation with Eq. 1) evaluations of our computational model. The purpose of compressed sensing is then to recover the sparsest signal, $\Psi\mathbf{c}$, that produces the measurements f. This can be written as an optimization problem as

$$\hat{\mathbf{c}} = \arg\min_{\mathbf{c}\in\mathbb{R}^N} \|\mathbf{c}\|_0 \text{ subject to } \mathbf{b} = \Phi\Psi\mathbf{c}, \tag{7}$$

where $\|\mathbf{c}\|_0$ is defined as the number of nonzero entries in $\mathbf{c}$. Finding a solution to this problem would require enumeration of all possibilities and is thus of combinatorial complexity. The fundamental insight in compressed sensing is the convex relaxation of Eq. 7 by using the l_1 norm to find the coefficients as

$$\hat{\mathbf{c}} = \arg\min_{\mathbf{c}\in\mathbb{R}^N} \|\mathbf{c}\|_1 \text{ subject to } \mathbf{b} = \Phi\Psi\mathbf{c}, \tag{8}$$

where, $\|\mathbf{c}\|_1 = \sum_{k=1}^{N} |\mathbf{c}_k|$. With enough measurements, if f is sparse in Ψ, then it can nearly always be reconstructed from $\mathbf{b}$ using Eq. 8, as $f \approx \Psi\hat{\mathbf{c}}$ [35]. Eq. 8 can be implemented as a linear program, for which many efficient solution algorithms exist.

Our goal then in this work is to assume a basis exists such that the subfunctions of a given function, f, can be represented sparsely in that basis. By assuming low effective superposition dimension, we will then interrogate the subfunctions of f by fixing all inputs that are not active in a given subfunction. Doing this allows us to sparsely reconstruct the subfunctions of f with very few evaluations of f per subfunction. If the function is of low effective superposition dimension, then only a small number of subfunctions must be reconstructed and then added together to provide a nearly exact surrogate representation of f. We develop our methodology for constructing such surrogates in the following section.

3 METHODOLOGY

In this work we assume we have a sparse representation of a given function f in the basis of Legendre polynomials. In Section 4, we present results for an analytical function that is defined as sparse in the basis of Legendre polynomials to demonstrate the effectiveness of the approach when this type of information is known. However, we also present results on a cooled gas turbine blade computational model that was treated as a black box in which the Legendre polynomial basis representation performs extremely well.

The Legendre polynomials in one-dimension, L_n, on $[-1,1]$, can be written as

$$L_n(x) = \frac{1}{2^n n!}\frac{d^n}{dx^n}\left[(x^2-1)^n\right], \tag{9}$$

according to Rodrigues' formula. These polynomials are orthonormal with respect to the uniform measure on $[-1,1]$. When attempting find a sparse representation of a function, f, using Eq. 8, we randomly sample a few points in the input space to generate a few random samples of the output space (this ensures good incoherence of Ψ and Φ). For the case of Legendre polynomials, sampling randomly from the Chebyshev distribution, rather than uniformly, produces better recovery of signal information [36, 37]. Thus, to construct Ψ, we sample in the input space with the Chebyshev distribution (also know as the arcsine distribution), whose probability density function is

$$p(x) = \frac{1}{\pi\sqrt{1-x^2}}, \tag{10}$$

with support $[-1,1]$. To ensure we can handle arbitrary intervals, $[a,b]$, with the Legendre polynomials, we sample a Chebyshev point and bijectively map it onto $[a,b]$ as

$$x \mapsto a + (b-a)\frac{x+1}{2}. \tag{11}$$

To construct a Legendre polynomial basis for a function of d inputs, with order 0 to $N_j - 1$ one-dimensional Legendre polynomials in each dimension, j, we take tensor products. Let $\mathscr{L}_j$ be a matrix whose rows are made up of the Legendre polynomials evaluated at given samples of x_j. That is, each row has a specific sample of x_j associated with it. If we have M_j samples and order 0 through $N_j - 1$ Legendre polynomials for input j, then $\mathscr{L}_j$ is $M_j \times N_j$. We can then construct the matrix Ψ as

$$\Psi = \bigotimes_{j=1}^{d} \mathscr{L}_j, \tag{12}$$

where $\otimes$ is the Kronecker product of the matrices, $\mathscr{L}_j$, and Ψ is $M \times N$, where $M = \prod_{j=1}^{d} M_j$ and $N = \prod_{j=1}^{d} N_j$.

As can be seen by the formula for M, the number of samples required grows combinatorially with the number of inputs. Thus, even just a few samples in each dimension can quickly become

computationally infeasible if d is large and we are creating a basis over the product space of the inputs using Eq. 12. However, if we assume f is approximately additive, then there is no need to create a basis over the product space of the inputs, since higher order interaction terms in the HDMR of the function will be negligible. For example, if we know that the superposition effective dimension of a function, f, is $d_s = 1$, then the function (which is then referred to as purely additive) can be written exactly as

$$f(\mathbf{x}) = f_0 + \sum_{j=1}^{d} f_j(x_j). \tag{13}$$

To construct a surrogate representation of an f defined by Eq. 13, we need only construct individual surrogates for each f_j. To do this in our compressed sensing context, we can define $\Psi_j = \mathscr{L}_j$ and solve Eq. 8 to arrive at a sparse surrogate representation,

$$\hat{f}_j \approx f_j + \tilde{f}_0, \tag{14}$$

of the subfunction associated with x_j and an offset, $\tilde{f}_0$. The surrogate estimates an offset because we must fix the inputs $\{x_l\}_{l \neq j}$ to nominal values. The HDMR found by using a nominal point is referred to as a cut-HDMR, as discussed in Ref. [26]. We use $\tilde{f}_0$ here rather than f_0 or $\hat{f}_0$ because the offset in this expression depends entirely on the nominal values (so it is not unique in the sense of having only one possible f_0) of the inputs chosen and it is not an approximation. This concept is shown notionally for a two input function in the left plot of Fig. 1, where the squares represent samples of x_1 with x_2 fixed to a nominal value to estimate f_1, and the circles represent samples of x_2 with x_1 fixed to a nominal value. In each dimension, our approach is to start with a small number of samples (e.g., 3), and solve Eq. 8. We then compute $\|\hat{c}_j\|_0$. We then randomly add another Chebyshev point, solve Eq. 8, and compute $\|\hat{c}_j\|_0$ again. Once this "l_0 norm" reaches equilibrium, we assume we have arrived at a good representation of f_j. This is a heuristic for determining how many samples are required for any given input and requires further study. Certainly if $\|\hat{c}_j\|_0 = N_j$, then we do not have sparsity in the basis representation and the representation itself is likely a poor approximation of the subfunction f_j plus an offset. If this conclusion is reached at this point, then at worst we have spent $\sum_{j=1}^{d} N_j$ function evaluations, which can be repurposed for another approximation approach based on random sampling or used as part of a Monte Carlo simulation. If our sampling heuristic in each dimension of an additive function suggests we have a good sparse representation of each subfunction, then we may construct our surrogate of f as

$$\hat{f}(\mathbf{x}) = \tilde{f}_0 + \sum_{j=1}^{d} \hat{f}_j - d\tilde{f}_0, \tag{15}$$

where the offset term, $\tilde{f}_0$ is computed by evaluating f at a nominal value of each input.

To determine how well our additive surrogate representation is approximating f, we conduct a validation exercise after each additional term in the surrogate HDMR is added. To do this, we randomly sample a small number of points from $\mathbf{x}$, (e.g., 10 points), and evaluate f for each of these points. None of these points is a part of the point set used to generate the surrogate representation. At these points we compute the difference between our surrogate representation and the true function, $e_i(\mathbf{x}^i) = f(\mathbf{x}^i) - \hat{f}(\mathbf{x}^i)$. We then use $\|\mathbf{e}\|_2$, as an indicator of how well the surrogate is approximating f. As a heuristic, if $\|\mathbf{e}\|_2$ is below a user specified threshold, we consider the approximation sufficient. The implications of this heuristic are a topic of future work.

If, after adding all subfunctions of one input, that is, all possible $\hat{f}_j$ terms, we are still not approximating f well in our validation set, then we continue on to two input subfunctions. This is assuming also that a sparse representation for each subfunction, f_j, has been found. Moving to subfunctions of two inputs requires the tensor product of the Ψ_j matrices. For example, for a subfunction approximation of $f_{j,l}$, we compute the matrix $\Psi_{j,l} = \Psi_j \otimes \Psi_l$. This requires a tensor product of the one-dimensional Chebyshev point sets for x_j and x_l, and subsequent evaluation of f at those points. A two-dimensional tensor product of 9 Chebyshev points in each dimension is shown in the right plot of Fig. 1. Here an original set of 18 points used to estimate two single input subfunctions would become 81 points. Thus, to ensure that only two-input subfunctions of likely significance in the approximation of f are incorporated, we use a greedy-based procedure for selecting which interaction terms to use. For this, we estimate the main effect sensitivity indices for each input x_j as

$$\hat{S}_j = \frac{\mathbb{V}(\hat{f}_j)}{\mathbb{V}(\hat{f})}. \tag{16}$$

These sensitivity indices estimate the approximate main effect contribution of input, x_j, and should be estimated using the anticipated distributions of each input that will be used in uncertainty propagation through the surrogate once its construction is complete. Experience has shown that inputs with large main effect indices also tend to be those inputs involved in significant interaction effects. Thus, we add two input subfunctions by starting

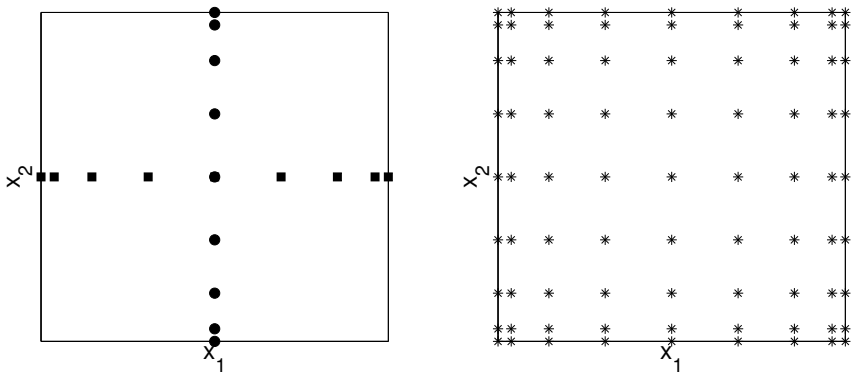

FIGURE 1. ONE DIMENSIONAL CHEBYSHEV DISTRIBUTED POINTS FOR TWO INPUTS FOR ESTIMATING ONE INPUT SUBFUNCTIONS (LEFT). TENSOR PRODUCT OF THE ONE DIMENSIONAL CHEBYSHEV DISTRIBUTED POINTS FOR ESTIMATING A TWO INPUT SUBFUNCTION (RIGHT).

with the input with the largest sensitivity index and the input with the second largest sensitivity index. If our heuristic error indicator has not fallen below the threshold after this subfunction is incorporated into the surrogate approximation, then we continue with the two input subfunction of the inputs with the largest and third largest sensitivity index, and continue in this manner until we have satisfied our error threshold, or determined it was wise to abandon the approximation in this particular basis representation. We note here that when two input subfunctions have been added to the surrogate representation, we must take care to avoid double-counting single input subfunction terms. That is, our two input subfunction approximation for inputs x_j and x_l is

$$\hat{f}_{j,l}(x_j,x_l) \approx f_{j,l}(x_j,x_l) + f_j(x_j) + f_l(x_l) + \tilde{f}_0. \tag{17}$$

Thus, if all possible two input subfunctions are added to the surrogate representation of f, we have

$$\hat{f}(\mathbf{x}) = \frac{(d-1)(d-2)}{2}\tilde{f}_0 - (d-2)\sum_{j=1}^{d}\hat{f}_j(x_j) + \sum_{\substack{j,l=1 \\ j<l}}^{d}\hat{f}_{j,l}(x_j,x_l). \tag{18}$$

Once we have a surrogate representation of $f(\mathbf{x})$ that we are satisfied with, we can proceed to propagate uncertainty through the surrogate using any supported distribution of $\mathbf{x}$. That is, any distribution whose probability density has values greater than zero only on the hypercube defined by the maps for each input defined by Eq. 11 and used in the surrogate construction. We note that our approach also provides approximations to the main effect sensitivity indices of each input, and variability of these sensitivities with respect to changing input distributions can rapidly be assessed.

4 RESULTS

We demonstrate our compressed sensing approach for uncertainty propagation on a two dimensional analytical function and on a finite element model of a cooled gas turbine blade. The analytical function is designed to show the effectiveness of our approach under perfect conditions. The cooled gas turbine blade model was treated as a black box, and thus, nothing was known about the sparsity of this model in the Legendre basis. This is the type of scenario we expect to encounter generally.

4.1 4-SPARSE LEGENDRE FUNCTION

The function we seek to develop a surrogate approximation for in this demonstration is

$$f(x_1,x_2) = 1 + x_1 + \frac{1}{2}(3x_2^2 - 1) + x_1x_2, \tag{19}$$

where $(x_1,x_2) \in [-1,1]^2$. This function has four modes in the Legendre basis spanned by the tensor product of the one dimensional Legendre bases for each individual dimension and no other components. Therefore, this function is known to be 4-sparse in the Legendre basis, and should be approximated exactly with very few function evaluations by our approach. We estimate the subfunctions, $\hat{f}_1$ and $\hat{f}_2$, as shown in Fig. 2. Here, the exact signal for each, (e.g., $f_1 + \tilde{f}_0$ for the x_1 component) is also plotted on each figure. The approximations of these subfunctions are exact, thus, only one curve is seen in each plot on the figure. In this case, $\tilde{f}_0$ is computed with $(x_1,x_2) = (0,0)$. The construction of each subfunction required 5 evaluations of

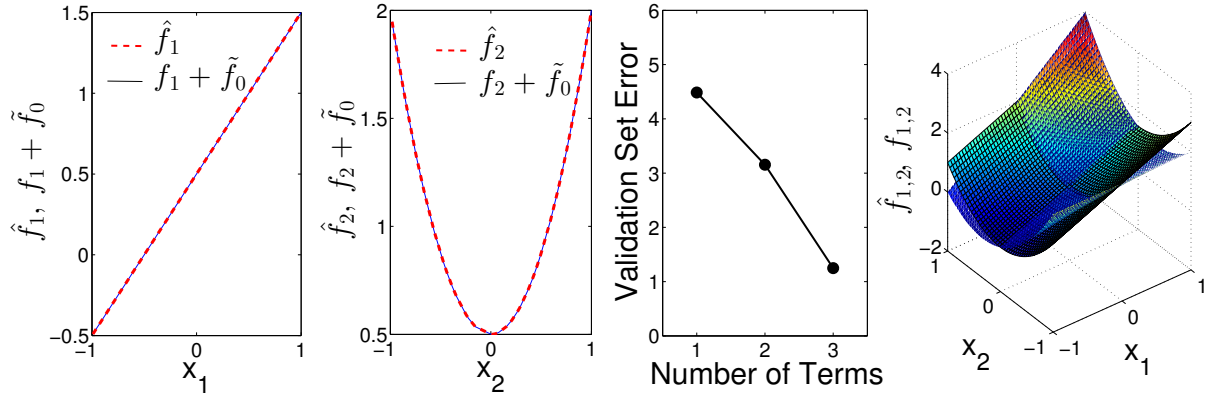

FIGURE 2. SUBFUNCTION OF INPUT X_1 AND ITS SPARSE APPROXIMATION (LEFT PLOT), SUBFUNCTION OF INPUT X_2 AND ITS SPARSE APPROXIMATION (SECOND FROM LEFT PLOT), VALIDATION ERROR AS A FUNCTION OF THE NUMBER OF TERMS IN THE SURROGATE APPROXIMATION (SECOND FROM THE RIGHT PLOT), AND THE FULL FUNCTION AND ITS ADDITIVE SURROGATE APPROXIMATION.

the function, $f(x_1,x_2)$ each. To determine this, we began with 2 samples for each subfunction construction independently and computed $\|\hat{c}_j\|_0$ for each input as described in Section 3. We continued with this process for 3, 4, and 5 samples, where it was revealed that the $\|\hat{c}_j\|_0$ terms for each input were no longer varying with increasing sample size. Thus, the process was stopped at 5 samples each. The third plot in the figure shows the validation error, $\|\mathbf{e}\|_2$, which was computed for each successive approximation with 10 randomly chosen values of (x_1,x_2) that were propagated through each approximation, as well as the full function, $f(x_1,x_2)$. The first error value is the error from using just $\tilde{f}_0$ to estimate $f(x_1,x_2)$. The second error value is associated with using $\hat{f}_1$, and the last error value is associated with using $\hat{f}_1+\hat{f}_2-\tilde{f}_0$. As can be seen from the plot, the error is decreasing as we add more terms to the approximation, but we still have not recovered the full signal from just the one input subfunctions. This can also be seen in the fourth plot of Fig. 2, where the functions $f_{1,2} = f(x_1,x_2)$ and $\hat{f}_{1,2}$ are plotted as a function of (x_1,x_2). The surfaces differ because the $\hat{f}_{1,2}$ function cannot account for the interaction effect of x_1 and x_2 in $f_{1,2}$ with only additive terms of one input subfunctions. We note that thus far, we have evaluated $f(x_1,x_2)$ a total of 20 times. These evaluations consist of 5 for each one input subfunction construction, and 10 for validation.

The next step in our modeling process then, is to add the two input subfunction of (x_1,x_2) following the procedure described in Section 3. The result of adding the two input subfunction to our original surrogate representation is shown in Fig. 3. The top plot overlays the surrogate approximation, $\hat{f}(x_1,x_2)$, and the full model, $f(x_1,x_2)$. The surrogate model is an exact match in this case. The bottom plot shows the cumulative distribution function (CDF) of $f(x_1,x_2)$, with $X_1 \sim U[0,1]$ and $X_2 \sim U[-.5,.5]$.

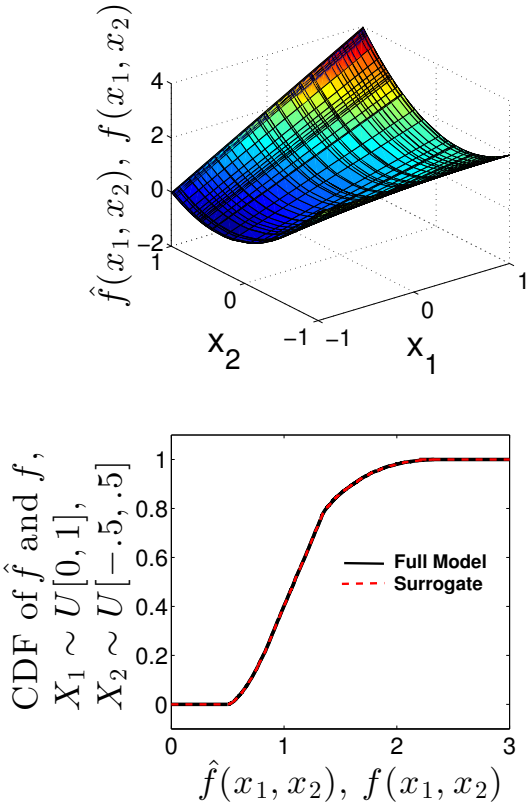

FIGURE 3. SURROGATE APPROXIMATION, $\hat{f}$, AND FULL MODEL, f (TOP). EMPIRICAL CDF FUNCTION OF FULL MODEL VIA MONTE CARLO SIMULATION (BLACK) AND SURROGATE APPROXIMATION (RED) (BOTTOM).

This CDF was estimated empirically via Monte Carlo simulation using 1000 full model evaluations and is shown as a black line. Also shown on this plot is the empirical CDF of $\hat{f}(x_1,x_2)$, which was estimated using 1000 samples of the surrogate approximation. This CDF is shown as a dashed red line and matches the true CDF exactly. To construct the surrogate approximation required only 25 full model evaluations, plus another 10 full model evaluations for validation. We use the tensor product of the previously computed sample points when constructing the two input subfunction, which allows us to reuse the 10 full model evaluations acquired in the construction of the one input subfunctions.

4.2 COOLED GAS TURBINE BLADE MODEL

For this demonstration we consider a finite element model of a cooled gas turbine blade modeled after that provided in Ref. [38]. The blade profile and the random inputs to the finite element model are shown in Fig. 4. The computational model is

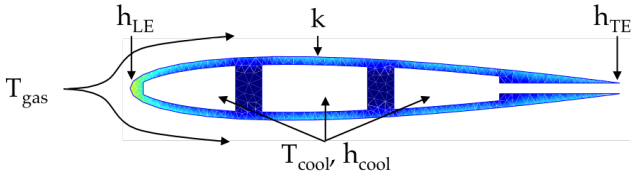

FIGURE 4. THE COOLED GAS TURBINE BLADE PROFILE AND RANDOM INPUT VARIABLES.

a heat transfer model that simulates a cooled gas turbine blade in a hot gas path flow. The uncertain inputs to this system, their units, and their probability distributions are provided in Tab. 1. Here, k is the thermal conductivity of the blade, T_{gas} is the external gas temperature, h_{LE} is the leading edge heat transfer coefficient, h_{TE} is the trailing edge heat transfer coefficient, T_{cool} is the cooling passage temperature, and h_{cool} is the cooling passage heat transfer coefficient. It is assumed that a failure will occur when the maximum temperature or temperature gradients become too large. This is modeled by combining damage due to high temperature and damage due to high temperature gradients as

$$D := \frac{T_{\max} - T_{\max}^{\text{des}}}{T_{\text{limit}} - T_{\max}^{\text{des}}} + \frac{|\nabla T|_{\max} - |\nabla T|_{\max}^{\text{des}}}{|\nabla T|_{\text{limit}} - |\nabla T|_{\max}^{\text{des}}}, \tag{20}$$

where $T_{\text{limit}} = 1,500K$ is the limiting value of the temperature, $|\nabla T|_{\text{limit}} = 80,000K/m$ is the limiting value of the temperature gradient, $T_{\max}^{\text{des}} = 1,430K$ is the maximum temperature at design-intent conditions, and $|\nabla T|_{\max}^{\text{des}} = 70,000K/m$ is the maximum temperature gradient at design-intent conditions. We consider

TABLE 1. COOLED GAS TURBINE BLADE MODEL INPUTS AND DISTRIBUTIONS, WHERE $\mathscr{T}(a,b,c)$ REPRESENTS A TRIANGULAR DISTRIBUTION WITH LOWER LIMIT, a, MODE, b, AND UPPER LIMIT, c.

Input	Units	Distribution
k	$W/(mK)$	$\mathscr{T}(28.5,30,31.5)$
T_{gas}	K	$\mathscr{T}(1400,1500,1600)$
h_{LE}	$W/(m^2K)$	$\mathscr{T}(15000,16000,17000)$
h_{TE}	$W/(m^2K)$	$\mathscr{T}(3500,4000,4500)$
T_{cool}	K	$\mathscr{T}(550,600,650)$
h_{cool}	$W/(m^2K)$	$\mathscr{T}(1400,1500,1600)$

the damage, D, as the quantity of interest of this system. Thus, our goal is to efficiently propagate uncertainty from the inputs to this particular output.

Using the methodology outlined in Section 3, we construct a compressed sensing surrogate representation of the damage output of the computational model of the cooled gas turbine blade. We construct this surrogate as an approximately additive function of the inputs using a sparse representation in the Legendre polynomial basis. After initially creating a purely additive surrogate (i.e., containing only the one input subfunctions), the approximate sensitivity indices, $\hat{S}_j$, for $j = 1,2,\ldots,6$, were computed according to Eq. 16. The results of this computation are presented in Tab. 2. According to our greedy strategy for incorporating higher order subfunctions, which was described in Section 3, we would first consider adding the two input subfunction of $(T_{\text{gas}}, T_{\text{cool}})$, followed by the two input subfunction of $(T_{\text{gas}}, h_{\text{cool}})$, and so on.

TABLE 2. APPROXIMATE MAIN EFFECT SENSITIVITY INDICES COMPUTED USING THE PURELY ADDITIVE SURROGATE MODEL.

Input	k	T_{gas}	h_{LE}	h_{TE}	T_{cool}	h_{cool}
$\hat{S}$	0.01	0.93	0.01	0.01	0.03	0.02

In Fig. 5, the results for uncertainty propagation through the surrogate representation containing all subfunctions of one variable and five subfunctions of two inputs, along with the results from Monte Carlo simulation with the full computational model are shown. The two input subfunctions included in the surrogate representation are the pairs of T_{gas} and each other variable. In Fig. 5, the top plot displays the histogram of Damage

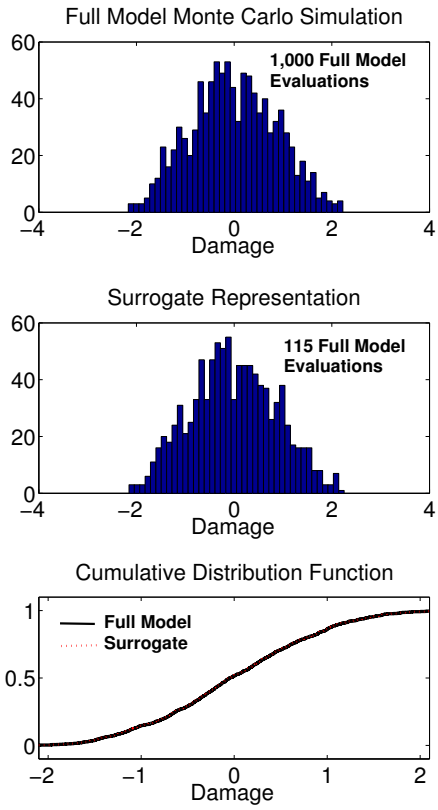

FIGURE 5. HISTOGRAM OF FULL MODEL MONTE CARLO SIMULATION (TOP). HISTOGRAM OF SURROGATE REPRESENTATION BASED MONTE CARLO SIMULATION (MIDDLE). EMPIRICAL CUMULATIVE DISTRIBUTION FUNCTIONS OF THE FULL MODEL AND SURROGATE REPRESENTATION (BOTTOM).

as computed by a 1,000 sample Monte Carlo simulation of the full model. The middle plot presents the histogram computed by sampling from the surrogate representation 1,000 times. The surrogate representation itself required only 115 full model evaluations, which consisted of 5 evaluations for each one input subfunction, 15 additional function evaluations for each two input subfunction included, and 10 full model evaluations for the validation set. As can be seen from the two histograms, the surrogate model provides an accurate representation of the full model. In the bottom plot of the figure, the empirical cumulative distribution function for the Monte Carlo simulation of the full model and the surrogate representation are plotted. As is clear from the plot, these functions are nearly identical.

While the results from the surrogate representation constructed from 115 points and 12 terms (the offset term, 6 one input subfunctions, and 5 two input subfunctions) are excellent and represent substantial computational savings, more could have been saved depending on the desired threshold for the validation error. As we did in the case of the analytical function in Section 4.1, we randomly sampled 10 points in the input space, propagated them through the full model, and used the results as a validation set to test against with our surrogate representations. The results of this test as we added more terms to our approximation are shown in Fig. 6. The figure clearly shows

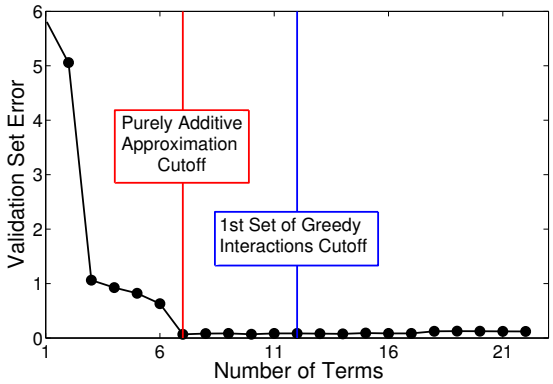

FIGURE 6. VALIDATION ERROR AS A FUNCTION OF THE NUMBER OF TERMS IN THE SURROGATE APPROXIMATION OF THE COOLED GAS TURBINE BLADE MODEL. THE RED LINE INDICATES THE END OF THE PURELY ADDITIVE SURROGATE TERMS AND THE BLUE LINE INDICATES THE END OF THE TERMS USED IN COMPUTED THE RESULTS SHOWN IN FIG. 5.

that the validation error did not decrease anymore after the first 7 terms (the offset and the one input subfunctions). Indeed, the results shown in Fig. 5 can be reproduced just as well with only the first 7 terms of the surrogate representation, which required only 40 full model evaluations to construct (including the 10 for the validation set). Had a reasonable threshold for the validation error been set, we would not have greedily added the two input subfunctions and the surrogate representation would have been essentially identical. The development of robust algorithms for constructing sparse surrogate representations that incorporate such heuristics is a topic for future work.

5 CONCLUSIONS AND FUTURE WORK

We have presented a compressed sensing approach to uncertainty propagation by constructing a sparse surrogate representation of a computational model. The approach relied on the assumption of approximate additivity. We demonstrated our approach on an approximately additive function that was known to be sparse in the Legendre basis and the results revealed the effec-

tiveness of our approach under perfect circumstances. We also demonstrated our approach on a finite element model of a cooled gas turbine blade for which we had no pre-requisite knowledge of sparsity in the Legendre basis or approximate additivity. Nevertheless, our approach yielded significant computational savings as compared to Monte Carlo simulation with negligible loss of accuracy in the uncertainty propagation results. We hypothesize that many other engineering computational models are approximately additive and can be represented well in a polynomial basis (or other bases as necessary). If this hypothesis is true, then the concept of using compressed sensing to efficiently find a sparse representation of the underlying function is promising. To ensure the exploitation of sparsity and approximate additivity in these situations, robust algorithms that identify a good basis to search for sparseness in, and when a good sparse representation has been found in that basis, are necessary. We have provided several heuristics in this work that have some potential for providing this robustness, and their study is a topic of future work.

ACKNOWLEDGMENT

This work was supported by the AFOSR MURI on multi-information sources of multi-physics systems under Award Number FA9550-15-1-0038, program manager Dr. Jean-Luc Cambier, and by AFOSR grant FA9550-11-1-0339 under the Dynamic Data-Driven Application Systems Program, program manager Dr. Frederica Darema.

REFERENCES

[1] Van der Waart, A., 1998. Asymptotic statistics, volume 27 of Cambridge Series in Statistical and Probabilistic Mathematics 03.

[2] Ma, J. Z., and Ackerman, E., 1993. "Parameter sensitivity of a model of viral epidemics simulated with Monte Carlo techniques. ii. durations and peaks". *International Journal of Bio-Medical Computing,* ***32***(3), pp. 255–268.

[3] MacDonald, R., Campbell, J., et al., 1986. "Valuation of supplemental and enhanced oil recovery projects with risk analysis". *Journal of Petroleum Technology,* ***38***(01), pp. 57–69.

[4] Eggert, R. J., 1995. "Design variation simulation of thick-walled cylinders". *Journal of Mechanical Design,* ***117***(2A), pp. 221–228.

[5] Rastegar, J., and Fardanesh, B., 1990. "Geometric synthesis of manipulators using the Monte Carlo method". *Journal of Mechanical Design,* ***112***(3), pp. 450–452.

[6] Amaral, S., Allaire, D., and Willcox, K., 2014. "A decomposition-based approach to uncertainty analysis of feed-forward multicomponent systems". *International Journal for Numerical Methods in Engineering,* ***100***(13), pp. 982–1005.

[7] Allaire, D., and Willcox, K., 2014. "Uncertainty assessment of complex models with application to aviation environmental policy-making". *Transport Policy,* ***34***, pp. 109–113.

[8] Allaire, D., Noel, G., Willcox, K., and Cointin, R., 2014. "Uncertainty quantification of an aviation environmental toolsuite". *Reliability Engineering & System Safety,* ***126***, pp. 14–24.

[9] Grimmett, G., and Stirzaker, D., 2001. *Probability and Random Processes*. Oxford University Press.

[10] Helton, J. C., and Davis, F. J., 2003. "Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems". *Reliability Engineering & System Safety,* ***81***(1), pp. 23–69.

[11] Niederreiter, H., 2010. *Quasi-Monte Carlo Methods*. Wiley Online Library.

[12] McKay, M. D., Beckman, R. J., and Conover, W. J., 2000. "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code". *Technometrics,* ***42***(1), pp. 55–61.

[13] Simpson, T. W., Lin, D. K., and Chen, W., 2001. "Sampling strategies for computer experiments: design and analysis". *International Journal of Reliability and Applications,* ***2***(3), pp. 209–240.

[14] Jin, R., Chen, W., and Sudjianto, A., 2002. "On sequential sampling for global metamodeling in engineering design". In ASME 2002 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, pp. 539–548.

[15] Kennedy, M. C., and O'Hagan, A., 2001. "Bayesian calibration of computer models". *Journal of the Royal Statistical Society: Series B (Statistical Methodology),* ***63***(3), pp. 425–464.

[16] Eldred, M., Giunta, A., Collis, S., Alexandrov, N., Lewis, R., et al., 2004. "Second-order corrections for surrogate-based optimization with model hierarchies". In Proceedings of the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, NY,, Aug, pp. 2013–2014.

[17] Allaire, D., and Willcox, K., 2010. "Surrogate modeling for uncertainty assessment with application to aviation environmental system models". *AIAA Journal,* ***48***(8), pp. 1791–1803.

[18] Antoulas, A. C., 2005. *Approximation of Large-scale Dynamical Systems*. SIAM.

[19] Yao, W., Chen, X., Luo, W., van Tooren, M., and Guo, J., 2011. "Review of uncertainty-based multidisciplinary design optimization methods for aerospace vehicles". *Progress in Aerospace Sciences,* ***47***(6), pp. 450–479.

[20] Gu, X. S., Renaud, J. E., and Penninger, C. L., 2006. "Implicit uncertainty propagation for robust collaborative

optimization". *Journal of Mechanical Design,* ***128***(4), pp. 1001–1013.

[21] McDonald, M., and Mahadevan, S., 2008. "Uncertainty quantification and propagation for multidisciplinary system analysis". In 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Victoria, BC, Canada, Sep, pp. 9–12.

[22] Kokkolaras, M., Mourelatos, Z. P., and Papalambros, P. Y., 2006. "Design optimization of hierarchically decomposed multilevel systems under uncertainty". *Journal of Mechanical Design,* ***128***(2), pp. 503–508.

[23] Mahadevan, S., and Smith, N., 2006. "Efficient first-order reliability analysis of multidisciplinary systems". *International Journal of Reliability and Safety,* ***1***(1-2), pp. 137–154.

[24] Julier, S. J., 2002. "The scaled unscented transformation". In American Control Conference, 2002. Proceedings of the 2002, Vol. 6, IEEE, pp. 4555–4559.

[25] Sobol, I. M., 2003. "Theorems and examples on high dimensional model representation". *Reliability Engineering & System Safety,* ***79***(2), pp. 187–193.

[26] Rabitz, H., and Aliş, Ö. F., 1999. "General foundations of high-dimensional model representations". *Journal of Mathematical Chemistry,* ***25***(2-3), pp. 197–233.

[27] Ma, X., and Zabaras, N., 2010. "An adaptive high-dimensional stochastic model representation technique for the solution of stochastic partial differential equations". *Journal of Computational Physics,* ***229***(10), pp. 3884–3915.

[28] Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., and Tarantola, S., 2008. *Global Sensitivity Analysis: The Primer*. John Wiley & Sons.

[29] Griebel, M., and Holtz, M., 2010. "Dimension-wise integration of high-dimensional functions with applications to finance". *Journal of Complexity,* ***26***(5), pp. 455–489.

[30] Wang, X., and Fang, K.-T., 2003. "The effective dimension and quasi-Monte Carlo integration". *Journal of Complexity,* ***19***(2), pp. 101–124.

[31] Wang, X., and Sloan, I. H., 2005. "Why are high-dimensional finance problems often of low effective dimension?". *SIAM Journal on Scientific Computing,* ***27***(1), pp. 159–183.

[32] Donoho, D. L., 2006. "Compressed sensing". *Information Theory, IEEE Transactions on,* ***52***(4), pp. 1289–1306.

[33] Candes, E. J., and Tao, T., 2006. "Near-optimal signal recovery from random projections: Universal encoding strategies?". *Information Theory, IEEE Transactions on,* ***52***(12), pp. 5406–5425.

[34] Duarte, M. F., and Eldar, Y. C., 2011. "Structured compressed sensing: From theory to applications". *Signal Processing, IEEE Transactions on,* ***59***(9), pp. 4053–4085.

[35] Candès, E. J., Romberg, J., and Tao, T., 2006. "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information". *Information Theory, IEEE Transactions on,* ***52***(2), pp. 489–509.

[36] Brutman, L., 1997. "Lebesgue functions for polynomial interpolation – a survey". *Annals of Numerical Mathematics,* ***4***, pp. 111–127.

[37] Rauhut, H., and Ward, R., 2012. "Sparse Legendre expansions via $l1$-minimization". *Journal of Approximation Theory,* ***164***(5), pp. 517–533.

[38] Darmofal, D., 2016. 16.901 Computational methods in aerospace engineering, spring 2005, accessed 27 Jan, 2016, MIT OpenCoursWare, http://ocw.mit.edu.