

Physics-Based Surrogate Modeling of Parameterized PDEs for Optimization and Uncertainty Analysis

Prasanth B. Nair*

University of Southampton, Highfield, Southampton SO17 1BJ, U.K

Abstract

This paper presents physics-based surrogate modeling algorithms for systems governed by parameterized partial differential equations (PDEs) commonly encountered in design optimization and uncertainty analysis. We first outline unsupervised learning approaches that leverage advances in the machine learning literature for a meshfree solution of PDEs. Subsequently, we propose continuum and discrete formulations for systems governed by parameterized steady-state PDEs. We consider the case of both deterministically and randomly parameterized systems. The basic idea is to embody the design variables or uncertain parameters in additional dimensions of the governing PDEs along with the spatial coordinates. We show that the undetermined parameters of the surrogate model can be estimated by minimizing a physics-based objective function derived using a multidimensional least-squares collocation or the Bubnov-Galerkin scheme. This potentially allows us to construct surrogate models without using data from computer experiments on a deterministic analysis code. Finally, we also outline an extension of the present approach to directly approximate the density function of random algebraic equations.

1 Introduction

Supervised learning is essentially concerned with the problem of discovering relationships in observational data $\mathcal{D} := (x_i, y(x_i)), i = 1, 2, \dots, m$, where $x_i \in \mathbb{R}^p$ denotes the input vector and $y \in \mathbb{R}$ is the target. The learning task may be interpreted as a function approximation or a classification problem depending on the case when the target y is a continuous variable or a class label, respectively. For a wide class of techniques, the model used to learn the input-output relationship can be written in the

general form $\hat{y}(x) = \sum_{j=1}^n \alpha_j k_j(x, \beta)$, * where α and β are undetermined parameters. $k(x, \beta)$ is a kernel, which is typically chosen to be a sigmoid or a radial basis function. In supervised learning, α and β are estimated by minimizing a loss function in terms of the observed target y . A detailed exposition of suitable loss functions such as least-squares, robust loss measures, and maximum likelihood can be found in the literature.¹⁻³

One popular application of supervised learning has been to construct surrogate models using data from computer experiments. A driving factor behind this has been the ever increasing need for computationally cheap approximation models for applications such as design optimization and uncertainty analysis, which require the analysis model to be evaluated repeatedly. This traditional approach involves running the analysis code for a number of inputs to generate a set of training/observational data. Subsequently, a surrogate model is trained to learn the input-output mapping using least-squares minimization techniques or Bayesian formalisms.³ Since, no information about the governing equations is employed, we shall refer to this supervised learning approach as black-box surrogate modeling.

A major drawback of supervised black-box modeling is that a large number of training points is required to construct an accurate surrogate. This is particularly true for problems with large number of input variables and highly nonlinear input-output relationships. This in turn leads to a significant increase in the computational cost due to the requirement of running the analysis code at a large number of design points. A number of recent studies^{4,5} have examined strategies to circumvent this *curse of dimensionality*, which arises from the fact that the number of hypercubes required to fill out a compact region of a p -dimensional space grows exponentially with p . Even though, promising results have been reported for some problems, the fundamental difficulties associated with black-box supervised learning

*Research Fellow, Computational Engineering and Design Center, School of Engineering Sciences, Member AIAA. P.B.Nair@soton.ac.uk

Copyright © 2002 by Prasanth B. Nair. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission.

*Examples include feedforward neural networks, radial basis networks, projection pursuit learning and support vector machines.

approaches are not expected to disappear.

In contrast, unsupervised learning approaches model the trends in the data without using the target y . In the machine learning and statistics literature, unsupervised learning is typically applied to exploratory data analysis and visualization; examples include Hopfield networks and Kohonen’s self organizing maps.⁶ In the context of function approximation, however, an unsupervised learning approach can be formulated only if it is possible to define a suitable loss function without using information about the target y . This would allow the model parameters α and β to be estimated via minimization of an objective function that indicates how well the model performs in the learning task.

In light of the foregoing discussion concerning supervised and unsupervised learning, it is instructive to consider a well-known PDE solution technique such as the finite element method. This technique may be considered to be an unsupervised learning approach since the field variables are approximated (using a surrogate model involving a linear combination of local basis functions) without using any observational data. Rather, the variational form of the governing equations is employed to estimate the model parameters.[†]

This interpretation of PDE solution techniques paves the way forward for the application of models other than grid-based local basis functions of the type used in finite element schemes. For example, radial basis functions and neural network models which share the attractive property of universal approximation can be employed as trial functions within a meshfree PDE solver. This idea was first proposed by Kansa,^{7,8} who investigated the application of radial basis functions to solve PDEs via collocation. Similarly, Lagaris et al.^{9–11} proposed using feedforward neural networks as trial functions for solving PDEs. It was shown that these meshfree schemes may give more accurate solutions than traditional grid-based algorithms. A review of recent research on similar meshfree schemes has been presented by Belytschko et al.¹²

In this paper, we consider the more general problem involving the numerical solution of PDEs in which the operators are parameterized in terms of deterministic or random variables. Our objective is to develop new unsupervised learning strategies to construct a physics-based surrogate model of the field variables. The proposed approach embodies the design variables or uncertain parameters in addi-

tional dimensions of the governing PDEs along with the spatial coordinates, i.e., we propose the interpretation of a parameterized PDE as a multidimensional operator problem with variable coefficients. We dub this approach *physics-based surrogate modeling*, since we estimate the model parameters by minimizing an objective function based on the equation governing the physics of the problem. In contrast, black-box approaches based on supervised learning rely exclusively on observational data - no attempt is made to ensure that the resulting surrogate satisfies the governing equations in some sense. It is worth noting that our approach is similar in spirit to reduced basis methods employing subspace projection schemes, which have been applied with a great deal of success to linear and nonlinear algebraic systems,^{13–15} eigenvalue problems,¹⁶ nonlinear CFD analysis,¹⁷ and linear stochastic PDEs.¹⁸

A key idea exploited in the present work is that a physics-based objective function can be defined for many applications where the governing equations as well as the numerical solution scheme are well understood. We show that the undetermined parameters of surrogate models such as neural networks or radial basis functions can be estimated by minimizing the physics-based objective function. The implication of this development is that a surrogate model can be constructed without using any observational data about the output or target to be approximated.

To illustrate the genesis of our approach, we first present an overview of the connection between the problem of learning from scattered observational data and that of solving PDEs. Subsequently, we consider the more general problem of solving parameterized steady-state PDEs. We propose two unsupervised approaches for surrogate modeling which fully exploit the physics of the system under consideration. The first approach can be interpreted as a meshfree numerical scheme for solving PDEs using parameter-dependent trial functions. The undetermined parameters of the surrogate model are estimated here by minimizing an objective function formulated using a multidimensional least-squares collocation or the Bubnov-Galerkin scheme. The boundary conditions are implemented via inequality constraints.

The second approach utilizes a discretized model of the governing PDEs to construct a surrogate model. The objective behind this is to reuse existing analysis capability without intrusive modifications to the source code, and also to alleviate the requirement of explicitly satisfying the boundary conditions. The structure of the surrogate model is chosen to be same as in the earlier approach. However, the undetermined parameters of the surrogate

[†]In the context of the finite element method, the nodal values of the field variables can be interpreted as the undetermined model parameters.

model are estimated by a modified least-squares collocation scheme, which minimizes the integral of the residual error in the discretized PDEs over the parameter space. We also outline avenues for extending the present approach to directly approximate the density function of random algebraic equations. Finally, numerical results are presented for some simple problems in structural mechanics.

2 Preliminaries

In this paper, we will focus on a class of learning models of the form $\hat{y}(x) = \sum_{j=1}^n \alpha_j k_j(x, \beta)$, where α and β are model parameters. This class of models includes radial basis and feedforward neural networks. The structure of radial basis kernels and their parameterization are shown in Table 1. The vector $c_i \in \mathbb{R}^p$ appearing within the kernel is commonly referred to as the center, which can be varied to create a family of kernels. The term β_i which appears in the parameterization of Gaussian kernels and Multiquadrics can be interpreted as a correlation parameter or a hyperparameter which governs the region of influence of the basis function in \mathbb{R}^p .

In the case of feedforward neural networks with a single hidden layer, the model may be similarly written as

$$\hat{y}(x) = \sum_{i=1}^n \alpha_i \sigma(a_i), \quad a_i = \sum_{j=1}^p w_{ij} x_j + \beta_j, \quad (1)$$

where α , w and β are model parameters representing the weights and bias terms of the network. $\sigma(x)$ is a transfer function which is typically chosen to be the sigmoid, i.e., $(1 + e^{-x})^{-1}$ or the hyperbolic tangent function. Note that the total number of undetermined parameters in the neural network is $(p+2)n$, where n and p denote the number of neurons in the hidden layer and the number of inputs, respectively.

Given a set of observational data $\mathcal{D} := (x_i, y(x_i))$, $i = 1, 2, \dots, m$, the learning task involves estimating the model parameters via minimization of a loss function. Consider the case when the least-squares error function is to be minimized. Then if the model parameters within the kernels are kept fixed, the vector α can be computed by solving a linear least-squares problem. In contrast, if the parameters within the kernel are considered to be unknown, it is necessary to solve a nonlinear least-squares problem. More recently, greedy approximation algorithms¹⁹ have been proposed to improve the computational efficiency and memory requirements by adaptively selecting the basis functions from an over-specified dictionary during the training phase. For a detailed

Table 1: Radial Basis Kernels

Nomenclature	$k_i(x, \beta)$
Linear Splines	$\ x - c_i\ $
Thin Plate Splines	$\ x - c_i\ \ln \ x - c_i\ $
Cubic Splines	$\ x - c_i\ ^3$
Gaussian	$e^{-\frac{\ x - c_i\ ^2}{\beta_i}}$
Multiquadrics	$\sqrt{1 + \frac{\ x - c_i\ ^2}{\beta_i}}$

exposition of the computational aspects of supervised learning the reader is referred to the excellent text by Bishop.²

Note that both radial basis and multilayer neural network models are theoretically capable of *universal approximation*,²⁰ i.e., they can approximate any function to an arbitrary degree of accuracy. In the next section, we briefly outline how these learning models can be employed in conjunction with classical collocation schemes to solve PDEs. For the sake of generality, we will use the notation $k(x, \beta)$ to denote basis functions which are parameter-free (such as linear, thin plate, and cubic splines) and also tunable kernels (e.g., the Gaussian function, Multiquadrics and those in feedforward neural networks).

3 Meshfree Solution of PDEs

To illustrate the application of unsupervised learning to numerical solution of PDEs, consider a multidimensional steady-state PDE of the form

$$\mathcal{L}u(x) = f(x) \quad \subset \Omega, \quad (2)$$

subject to the boundary conditions (BCs)

$$\mathcal{B}u(x) = g(x) \quad \subset \partial\Omega, \quad (3)$$

where \mathcal{L} and \mathcal{B} are differential operators in space $x \in \mathbb{R}^d$, and $u(x)$ denotes the field variable. We denote the computational domain and the boundary using the symbols Ω and $\partial\Omega$, respectively.

Meshfree collocation algorithms use a set of nodes situated within the domain as well as on the boundary,²¹ i.e.,

$$\mathcal{C} = \{(x_i)|_{i=1, n_d} \subset \Omega, (x_i)|_{i=n_d+1, n_d+n_b} \subset \partial\Omega\}, \quad (4)$$

where n_d and n_b denote the number of collocation points on the domain and the boundary, respectively.

Let us represent the field variable using learning models capable of universal approximation as trial

functions, i.e.,

$$\hat{u}(x) = \sum_{i=1}^n \alpha_i k_i(x, \beta) \quad (5)$$

To compute the undetermined coefficients α and β in (5), the governing equations can be collocated on the set \mathcal{C} to arrive at the following system of equations

$$\begin{aligned} \mathcal{L}u(x_i) &= \mathcal{L} \sum_{j=1}^n \alpha_j k_j(x_i) = f(x_i), i = 1, \dots, n_d \quad (6) \\ \mathcal{B}u(x_i) &= \mathcal{B} \sum_{j=1}^n \alpha_j k_j(x_i) \\ &= g(x_i), i = n_d + 1, \dots, n_d + n_b \quad (7) \end{aligned}$$

From the preceding equations, the connection between collocation schemes and the problem of supervised learning from scattered data may become clear to the reader. If we replace the operator \mathcal{L} with the identity and $f(x_i)$ with the target, the collocation principle becomes equivalent to that of supervised learning from scattered input-output data. This is the fundamental observation which motivates us to leverage recent advances in machine learning to develop new numerical schemes for solving PDEs.

Now consider the case when the kernel parameters, i.e., the centers c_i and the parameter β are considered to be known.[‡] Then for linear steady-state problems, if n is chosen to be equal to $n_d + n_b$, a square linear algebraic system of equations of the form $\tilde{K}\alpha = \tilde{f}$ can be obtained. For the cases when $n < n_d + n_b$ and $n > n_d + n_b$, an under-determined or over-determined linear least-squares problem has to be solved, respectively. It is worth noting that this least-squares problem can be efficiently solved by employing the adaptive greedy approximation framework proposed in Nair et al.¹⁹ Interestingly, such an approach would also allow the optimal set of collocation points to be adaptively selected during the solution process. For the case of nonlinear problems, similar iterative methods can be employed to solve the collocation system of equations.

An alternative approach has to be employed for the case when the parameters within the kernel are considered to be unknown. Consider, for example, the case when a feedforward neural network model is used to approximate the field variable $u(x)$. Then the following constrained optimization problem has to be solved to estimate the model parameters.

$$\text{Minimize} : \sum_{i=1}^{n_d} (\mathcal{L} \sum_{j=1}^n \alpha_j k_j(x_i) - f(x_i))^2 \quad (8)$$

[‡]In practice, the centers c_i can be chosen to coincide with the set of collocation points. The parameter β is either set to a sensible constant or $\min_{j \neq i} \|x_i - x_j\|^2$ for the i th basis.

$$\text{Subject to} : \sum_{i=n_d+1}^{n_d+n_b} (\mathcal{B} \sum_{j=1}^n \alpha_j k_j(x_i) - g(x_i))^2 \leq \epsilon, \quad (9)$$

where ϵ is a small parameter which indicates the extent to which the BCs are to be satisfied.

Lagaris et al.¹⁰ have examined in detail procedures for handling the BCs in neural network approaches to solving PDEs. They proposed the use of a hybrid approach wherein the neural network is combined with radial basis functions to ensure that the BCs are properly satisfied. Extremely encouraging results were presented for the solution of a class of linear as well as nonlinear PDEs on geometrically complex computational domains. An overview of other meshfree techniques such as the element-free Galerkin method and Petrov-Galerkin schemes can be found elsewhere.¹² Issues involved in applying meshfree algorithms to time-dependent PDEs have been presented by Kansa.^{7,8}

In the next section, we consider the more complex scenario when the PDE operator \mathcal{L} is parameterized in terms of a number of variables. These variables are considered to be either deterministic but bounded within an interval, or random in nature with a specified probability density function (pdf).

4 A Continuum Formulation for Physics-Based Surrogate Modeling

In this section, we present a continuum formulation for solving PDEs which are parameterized in terms of deterministic or random variables. To illustrate, let us consider a model steady-state parameterized PDE of the form

$$\mathcal{L}(x, \theta)u = f(x, \theta), \quad (10)$$

subject to the BCs

$$\mathcal{B}(x, \theta)u = g(x, \theta), \quad (11)$$

where $x \in \mathbb{R}^d$ denotes the spatial coordinates, $\theta \in \mathbb{R}^p$ denotes the vector of bounded design or random variables, and $u \in \mathbb{R}$ is the field variable to be computed. $\mathcal{L}(x, \theta)$ and $\mathcal{B}(x, \theta)$ denote parameterized operators.

Let $\Omega_x \in \mathbb{R}^d$ and $\Omega_\theta \in \mathbb{R}^p$ denote the domains over which the spatial coordinates and the parameter vector θ are defined, respectively. For the case of deterministic parameterization, Ω_θ can be interpreted as a bounded box formed using the bound constraints for each element of θ .

Let us now define what we mean by a solution of the governing parameterized equations in (10-11). It can be readily noted that the field variable is a function of x and θ . Hence, we may define the solution $u(x, \theta)$ as a model which can be employed to compute the field variable u for any arbitrary value of x and θ in $\Omega_x \cup \Omega_\theta$. This definition of the solution naturally suggests the interpretation of a parameterized PDE as a multidimensional problem,[§] i.e., we consider the vector θ as an additional set of coordinates along with x .

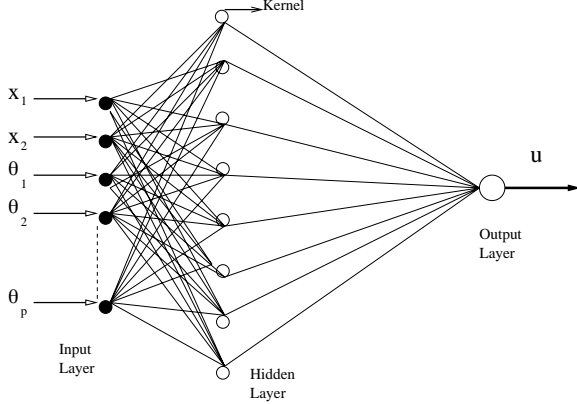


Figure 1: The field variable model in terms of the spatial coordinates x_1, x_2 and the parameter vector θ for a typical 2D parameterized PDE.

Clearly, from previous experiences with multidimensional PDEs,²⁴ it is well known that grid-based methods can be highly inefficient due to the curse of dimensionality. Our objective is to circumvent this difficulty by employing meshfree schemes which tackle multidimensional problems more elegantly. In particular, we propose to solve the governing PDE in (10,11) using parameter-dependent trial functions, i.e., a model of the form

$$\hat{u}(x, \theta) = \sum_{i=1}^n \alpha_i k_i(x, \theta, \beta), \quad (12)$$

where the kernel k_i is a function of both the spatial coordinates and the vector θ . The general structure of this model for a typical 2D steady-state PDE is graphically shown in Figure 1.

To compute the undetermined coefficient vector α and the kernel hyperparameters, one may use either a Bubnov-Galerkin scheme[¶] or a least-squares multidimensional collocation procedure. Let us first

[§]Strictly speaking, we interpret a parameterized PDE as a multidimensional operator problem with variable coefficients.

[¶]It may also be noted that other variational principles such as the Petrov-Galerkin scheme can be applied to this problem.

consider a least-squares collocation procedure to estimate the parameters in (12). This involves the solution of the following constrained optimization problem to estimate α and β .

$$\text{Minimize : } \int_{\Omega_x} \int_{\Omega_\theta} R_d^2 dx p(\theta) d\theta \quad (13)$$

$$\text{Subject to : } \int_{\Omega_x} \int_{\Omega_\theta} R_b^2 dx p(\theta) d\theta \leq \epsilon. \quad (14)$$

The functions R_d and R_b (which enforce the satisfaction of the the governing equations on the domain and the boundary, respectively) are defined below as

$$R_d = \mathcal{L}(x, \theta) \sum_{j=1}^n \alpha_j k_j(x, \theta, \beta) - f(x, \theta) \quad (15)$$

$$R_b = \mathcal{B}(x, \theta) \sum_{j=1}^n \alpha_j k_j(x, \theta, \beta) - g(x, \theta) \quad (16)$$

The function $p(\theta)$ in (13,14) is chosen to be unity for the case when the elements of θ are bounded deterministic variables. For the case when the elements of θ are random variables, $p(\theta)$ denotes the joint pdf.

In the Bubnov-Galerkin scheme, it is desired to estimate the model parameters in (12) such that the residual errors are orthogonal to the approximating space (i.e., $k_j(x, \theta, \beta)$). Hence, the objective function in (13) can be rewritten as

$$\sum_{j=1}^n \left(\int_{\Omega_x} \int_{\Omega_\theta} k_j(x, \theta, \beta) R_d dx p(\theta) d\theta \right)^2 \quad (17)$$

Similarly, the constraint in (14) becomes

$$\sum_{j=1}^n \left(\int_{\Omega_x} \int_{\Omega_\theta} k_j(x, \theta, \beta) R_b dx p(\theta) d\theta \right)^2 \quad (18)$$

Note that when the elements of θ are random variables, the preceding integrals essentially use the probability measure $p(\theta) d\theta$ to enforce orthogonality. However, for the case of deterministic parameterization since we set $p(\theta) = 1$, the probability measure reduces to the standard Lebesgue measure.

It can be seen from (15,16) that the multidimensional integrals for R_d and R_b cannot be simplified except for some special cases.^{||} In the present research, we employ space-filling experimental design techniques to numerically compute the integrals. The term *space-filling* is used here to indicate

^{||}For example, when basis functions obeying the product correlation rule²³ are applied to linear steady-state PDEs, it can be shown that the multidimensional integral collapses into a product of 1D integrals.

techniques which generate points by maximizing the distance function $\min \|c^i - c^j\|, \forall i \neq j$. There exists a wealth of techniques in the literature to generate such experimental designs. In our current numerical implementation, we use Latin hypercube sampling techniques to choose the collocation points.

In order to solve the nonlinear programming problem efficiently, it is required to compute the sensitivities of R_d and R_b with respect to the undetermined parameters in (12). For example, the sensitivities of R_d with respect to α_i may be evaluated as

$$\frac{\partial R_d}{\partial \alpha_i} = \frac{\partial \mathcal{L}(x, \theta, u)}{\partial u} \frac{\partial \hat{u}}{\partial \alpha_i} \quad (19)$$

The term $\partial \hat{u} / \partial \alpha_i$ can be readily computed by differentiating (12). This essentially entails computing the sensitivities of a learning model with respect to its inputs. A detailed exposition of computationally efficient techniques for sensitivity analysis (particularly for neural network models) can be found in standard texts; see, for example, Bishop.⁷ The sensitivities of R_b can be computed in a similar fashion.

4.1 Remarks

It is worth noting that the constrained nonlinear optimization problem reduces to a linear least-squares problem if the operators \mathcal{L} and \mathcal{B} are linear and a fixed radial basis kernel is chosen in (12). Even though the continuum approach is conceptually appealing, there are a number of problematic issues facing its application in practice. One major concern is how well the model will satisfy the BCs, particularly for cases when θ includes geometric parameters. From a theoretical point of view, one can appeal to the ability of universal approximation of the learning models considered here. This implies that given sufficient number of basis functions (n) in (12), the field variables can be approximated to an arbitrary degree of accuracy. Another possible criticism of the continuum approach is that it does not leverage existing analysis capability - rather it is necessary to write a new multidimensional solver. In the next section, we present a discrete formulation for physics-based surrogate modeling that alleviates some of these concerns.

5 A Discrete Formulation for Physics-Based Surrogate Modeling

In this section, we present a discrete formulation which directly deals with the spatially discretized version of the governing PDEs in (10,11). In this approach, we use a traditional analysis scheme such

as the finite element method for spatial discretization as well as to satisfy the BCs. The motivation for this is to reuse existing analysis software and solve an unconstrained optimization problem in contrast to the more complex nonlinear programming problem obtained for the continuum approach.

To illustrate the discrete formulation, consider a 2D nonlinear steady-state PDE with one field variable u . Further, let the governing PDE be discretized using finite elements (FE), where each node has two degrees of freedom u_1 and u_2 . The spatially discretized version of the governing PDE after the incorporating the BCs may then be written as

$$\mathbf{R}(\mathbf{w}, \theta) = 0, \quad (20)$$

where $\mathbf{w} \in \mathbb{R}^{2q}$ denotes the vector of discretized field variables at the q unconstrained nodes of the FE mesh. $\theta \in \mathbb{R}^p$ is the vector of parameters which are considered to be either deterministic or random variables. Note that \mathbf{w} may also be written as

$$\mathbf{w} = \{u_1(x_1, y_1, \theta), u_2(x_1, y_1, \theta), \dots, u_1(x_q, y_q, \theta), u_2(x_q, y_q, \theta)\}, \quad (21)$$

where x_i, y_i $i = 1, 2, \dots, q$ are the x and y coordinates of the q nodes in the FE mesh.

In contrast to the continuum approach, we propose to approximate the discretized field variables at the unconstrained nodes of the FE mesh. For the example 2D problem under consideration, we approximate u_1 and u_2 using models of the form

$$\hat{u}_1(x, y, \theta) = \sum_{i=1}^n \alpha_i^1 k_i(x, y, \theta, \beta_1), \quad (22)$$

$$\hat{u}_2(x, y, \theta) = \sum_{i=1}^n \alpha_i^2 k_i(x, y, \theta, \beta_1), \quad (23)$$

As mentioned earlier, in the discrete approach, we estimate the model parameters in (22,23) such that the discretized governing equations are satisfied in some sense. Objective functions which enforce satisfaction of (20) may be derived either using the Bubnov-Galerkin scheme or a multidimensional least-squares collocation procedure. In the latter case, the following unconstrained optimization problem has to be solved to estimate the model parameters in (22,23).

$$\underset{\alpha^1, \alpha^2, \beta_1, \beta_2}{\text{Minimize}} J(\mathbf{w}) = \int_{\Omega_\theta} \|\mathbf{R}(\hat{\mathbf{w}}, \theta)\| p(\theta) d\theta, \quad (24)$$

where $\hat{\mathbf{w}}$ is the vector formed by substituting the models (22,23) in (21).

When the Bubnov-Galerkin is employed, we seek to compute the undetermined parameters in (22,23) such that the residual error vector $\mathbf{R}(\mathbf{w}, \theta)$ is orthogonal to the approximating space. This leads us to the following unconstrained optimization problem

$$\text{Minimize}_{\alpha^1, \alpha^2, \beta_1, \beta_2} \sum_{i=1}^n \left(\int_{\Omega_\theta} \phi_i(\theta)^T \mathbf{R}(\hat{\mathbf{w}}, \theta) p(\theta) d\theta \right)^2, \quad (25)$$

where

$$\begin{aligned} \phi_i(\theta)^T &= \{k_i(x_1, y_1, \theta, \beta_1), k_i(x_1, y_1, \theta, \beta_2), \dots, \\ &\dots k_i(x_q, y_q, \theta, \beta_1), k_i(x_q, y_q, \theta, \beta_2)\} \in \mathbb{R}^{2q} \end{aligned} \quad (26)$$

Similar to the continuum formulation, the probability measure $p(\theta)d\theta$ reduces to the Lebesgue measure when the elements of θ are deterministic.

It can be seen that in contrast to the continuum approach, we end up with a much simpler unconstrained nonlinear optimization problem.** The minimization of J may be carried out either using evolutionary algorithms or gradient-based nonlinear programming techniques. Further, if the Jacobian matrix $\partial\mathbf{R}/\partial\mathbf{w}$ is available from the existing analysis code, then $\partial J/\partial\alpha$, can be efficiently computed since $\partial\mathbf{R}/\partial\mathbf{w} = \partial\mathbf{R}/\partial\mathbf{w} \times \partial\mathbf{w}/\partial\alpha$. Recollect that $\partial\mathbf{w}/\partial\alpha$ can be readily calculated by differentiating the models in (22,23) and using (21).

5.1 Implementation Issues

In this subsection, we briefly discuss how the discrete formulation can be implemented without any intrusive modifications to an existing analysis code which performs spatial discretization of the governing PDEs. We assume that - (1) a automatic mesh generation/deformation tool is available to generate meshes for various values of θ , (2) the vertices of the resulting mesh (x_i, y_i) are available, and (3) the analysis code when given the values of the field variables at the nodes of the mesh and θ returns the value of the residual error $\|\mathbf{R}\|$. Note that these assumptions are satisfied by many commercially available FE software and in-house PDE solvers.

For the sake of illustration, consider the case when a neural network is employed to model u_1 and u_2 using the least-squares collocation procedure. The steps involved in computing the objective function J for this case are summarized below:

**For certain choices of the kernel and linear PDE operators, it can be shown that the solution of the unconstrained optimization problem can be computed by solving a linear least-squares problem.

- Initialize the values of the model parameters in (22,23). Generate k samples of θ using Latin hypercube sampling, say $[\theta^1, \theta^2, \dots, \theta^k]$.

- Set $J = 0$. For each θ^i ,

DO for $i = 1, k$

- Generate FE mesh and extract the nodes of the mesh (x_j, y_j) .

- Use the learning models to generate the field variables u_1 and u_2 at all the nodes of the mesh and construct $\hat{\mathbf{w}}$.

- Call the analysis code with $\hat{\mathbf{w}}$ and θ^i , and request it to return the value of the residual $\|\mathbf{R}_i(\hat{\mathbf{w}}, \theta^i)\|$.

- $J = J + \|\mathbf{R}_i\|$.

END DO

It can be clearly seen from the above procedure that an attractive feature of the present approach is the possibility of massive parallelism since $\|\mathbf{R}_i\|$ can be computed independently for each realization of θ . A second level of parallelization arises from the fact that for many governing equations such as the Euler and Navier-Stokes equations, $\|\mathbf{R}\|$ may be computed in parallel.

6 Towards Density Estimation for Random Equations

In the earlier sections, we proposed to solve randomly parameterized PDEs by approximating the solution process as a function of the spatial coordinates and random parameters. In this section, we explore the possibility of extending the present approach to directly approximate the density function of the solution process. In other words, we consider the following problem:

Problem (P): Given the statistical moments (or the pdf $p(\theta)$) of $\theta \in \mathbb{R}^p$ and vectors of functions $A(u, \theta)$ and $b(\theta) \in \mathbb{R}^n$, find the joint pdf of $u \in \mathbb{R}^n$ and θ , when u satisfies the following system of random algebraic equations.

$$A(u, \theta) = b(\theta) \quad (P)$$

It is straightforward to list a set of constraints that the joint density function $p(u, \theta)$ must satisfy. One fundamental constraint is that $p(u, \theta)$ must be a *bonafide* density function, i.e.,

$$\int_{-\infty}^{+\infty} p(u, \theta) du d\theta = 1.0, \text{ and } p(u, \theta) \geq 0 \quad (27)$$

Further, since $p(\theta)$ is assumed to be given, the following equation must be valid

$$\int_{-\infty}^{+\infty} p(u, \theta) du = p(\theta), \quad (28)$$

However, in practice, when the statistics of θ are estimated from limited samples obtained via physical experimentation, only the first few statistical moments may be available. In such cases, (28) has to be substituted with the set of conditions.

$$\int_{-\infty}^{+\infty} \theta_i^k p(u, \theta) dud\theta = \langle \theta_i^k \rangle, i = 1, 2, \dots, p, \quad (29)$$

So far, we have only listed the trivial constraints that the desired solution $p(u, \theta)$ must satisfy. To proceed further, we need to rigorously define what it means to solve (P). We now introduce the notions of strong and weak solutions to (P) and see how far we can go from there.

Definition 1: $p(u, \theta)$ is called a strong solution of (P), iff for any realization u_i and θ_i drawn from $p(u, \theta)$, $A(u, \theta) = b(\theta)$ holds with probability one.

Definition 2: $p(u, \theta)$ is called a weak solution of (P) in the \mathcal{L}_q sense, if

$$\int_{-\infty}^{+\infty} \|A(u, \theta) - b(\theta)\|_q p(u, \theta) dud\theta = 0 \quad (30)$$

The notion of strong solution is rather stringent and may, in practice, be impossible to meet. It can be noted that Definition 2 is motivated by the physics-based objective function defined earlier in (24). We now examine whether a numerical scheme can be developed for approximating $p(u, \theta)$ by enforcing the notion of weak solution in the \mathcal{L}_q sense.

6.1 A Variational Formulation for $p(u, \theta)$

At the time of writing this paper, it is not clear whether the conditions mentioned earlier are sufficient to ensure the well-posedness of the density estimation problem. Hence, it makes sense to introduce regularization (such as Tikhonov regularization) to ensure that a meaningful solution is obtained. An alternative approach would be to employ the concept of maximum information entropy²² to ensure that the computed solution $p(u, \theta)$ is honest, i.e., we choose the solution which is maximally uncertain given that the constraint equations are satisfied. The constrained entropy maximization formulation based on Shannon's entropy measure can be stated as a variational problem of the form:

$$\text{Maximize : } \int_{-\infty}^{+\infty} -p(u, \theta) \ln p(u, \theta) dud\theta \quad (31)$$

Subject to:

$$\int_{-\infty}^{+\infty} \|A(u, \theta) - b(\theta)\|_q p(u, \theta) dud\theta = 0 \quad (32)$$

$$\int_{-\infty}^{+\infty} h_i(u, \theta) p(u, \theta) dud\theta = \gamma_i, i = 1, 2, \dots, l \quad (33)$$

where (33) is a generalized representation of the constraints mentioned earlier in (27-29).

Introducing a set of Lagrange multipliers ($\lambda_i, i = 0, 1, \dots, p$), the solution for $p(u, \theta)$ turns out to be

$$p(u, \theta) = a_0 e^{-\lambda_0 \|A(u, \theta) - b(\theta)\|_q + \sum_{i=1}^l \lambda_i h_i}, \quad (34)$$

where a_0 is an undetermined parameter to ensure normalization. Conceptually, the Lagrange multipliers can be computed by substituting (34) back into the constraint equations and solving a deterministic nonlinear system of equations. Unfortunately, the final solution (34) does not obey the product correlation rule, due to which computation of the Lagrange multipliers involves evaluation of multidimensional integrals. In the next subsection, we examine a numerically tractable procedure which results when a kernel expansion is used for approximating $p(u, \theta)$.

6.2 Numerical Approximation of $p(u, \theta)$

Consider the case when we wish to approximate the weak solution of (P) in the \mathcal{L}_2 sense given the statistics of θ . In order to avoid dealing with a variational problem, let us approximate $p(u, \theta)$ using the expansion

$$\hat{p}(u, \theta) = \sum_{i=1}^m \alpha_i k_i(u, \theta), \quad (35)$$

where k_i is the Gaussian kernel which obeys the product correlation rule. This restriction ensures that \hat{p} will be a *bonafide* density, and further the multidimensional integrals collapse into a product of one-dimensional integrals.

To compute the undetermined vector $\alpha \in \mathbf{R}^m$, we use the following equations:

$$\sum_{i=1}^m \alpha_i \int_{-\infty}^{+\infty} \|A(u, \theta) - b(\theta)\|_2 k_i(u, \theta) dud\theta = 0 \quad (36)$$

$$\sum_{i=1}^m \alpha_i = 1 \quad (37)$$

$$\sum_{i=1}^m \alpha_i \int_{-\infty}^{+\infty} h_j(u, \theta) k_i(u, \theta) dud\theta = \gamma_j, j = 1, \dots, l \quad (38)$$

It can be noted that the above equations can be written as a system of under-determined linear equations for α , when $m > l + 2$. Hence, conventional Tikhonov regularization¹ can be readily employed to ensure well-posedness. It is also worth noting that the formulations presented here can be applied even when a complete statistical characterization of θ is not available.

7 Examples

We illustrate the application of the present approach to two simple example problems from structural mechanics. The first problem shown in Figure 2 involves a stepped cantilevered rod subject to a axial load. We choose this problem since it corresponds to a simple 1D problem for which graphical visualization of the performance is readily possible. In the physics-based surrogate modeling approach, we seek to approximate the axial displacement u as a function of the coordinate x and the variables A_1 and A_2 . We consider the case when both these latter variables lie in the box $[0.5, 1.5]$.

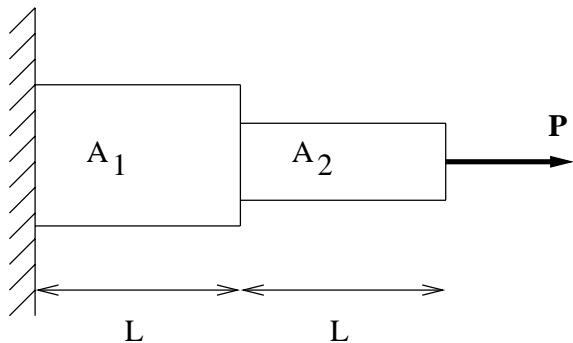


Figure 2: A stepped cantilevered rod with two design variables. $EA_0 = 6.987 \times 10^{-6}$, $L = 0.5$, and $\mathbf{P} = 10000$.

We use a feedforward neural network with 10 nodes in the hidden layer to model the relationship $u(x, A_1, A_2)$. Due to ease of implementation, we applied the discrete formulation to this test problem. This involved the use of a standard finite element procedure with two elements to discretize the governing differential equations in x to arrive at a system of linear algebraic equations similar to (20). The objective function was computed using a Latin hypercube with 30 samples. Recollect that the sampling procedure is applied only to evaluate the multidimensional integral in (24); in contrast to supervised black-box surrogate modeling, we do not use any samples of the field variable u . To estimate the neural network parameters, we used a genetic algorithm search followed by a BFGS optimizer.

Figure 3 shows the residual error in the discretized

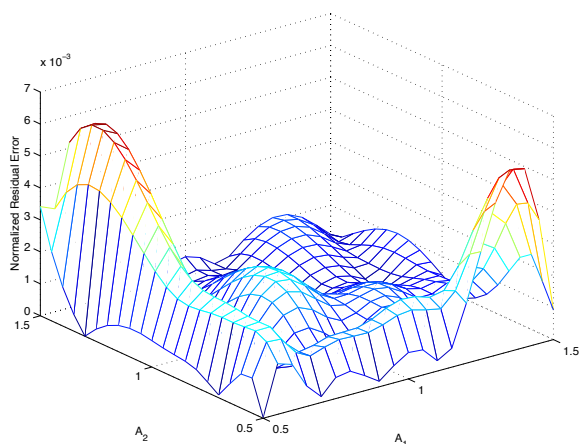


Figure 3: Residual Error in the discretized governing equations as a function of A_1 and A_2

governing equations for the rod problem as a function of the parameters A_1 and A_2 . Recollect that this error can be computed as $\|K(A_1, A_2)\hat{\mathbf{w}} - P\|$, where K is the stiffness matrix, P is the force vector, $\hat{\mathbf{w}} = \{\hat{u}(x_1, A_1, A_2), \hat{u}(x_2, A_1, A_2)\}^T$, and $x_1 = 0.5$ and $x_2 = 1.0$ are the x -coordinates of the two nodes. This residual error may be interpreted as an *a posteriori* error estimate which tells us how well the learning model (in this case a neural network with 10 nodes in the hidden layer) approximates the true value of the field variables. In practice, this information may be employed to decide the complexity of the learning model (i.e., the number of terms n) required to model the underlying relationship.

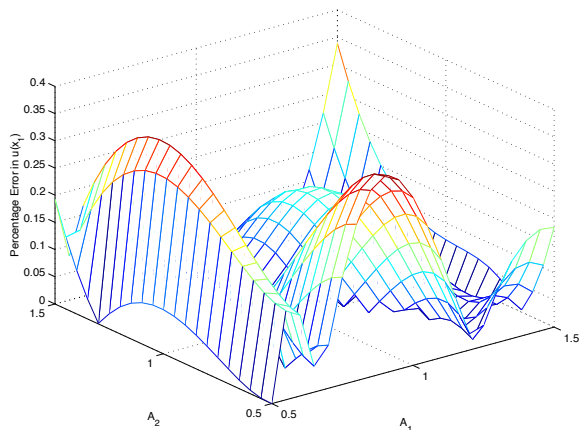


Figure 4: Percentage Error in $u(x_1)$ as a function of A_1 and A_2

The axial displacements at x_1 and x_2 were predicted using the trained physics-based surrogate model for a range of values of A_1 and A_2 . The percentage error in the axial displacements at x_1 and x_2

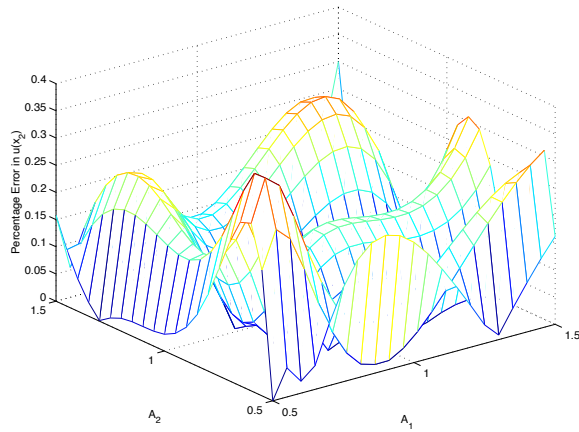


Figure 5: Percentage Error in $u(x_2)$ as a function of A_1 and A_2

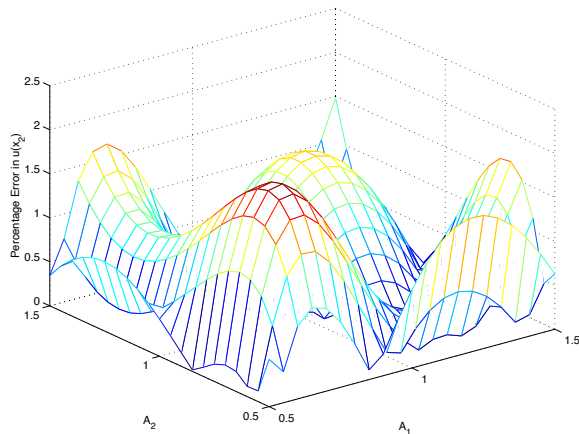


Figure 6: Percentage Error in the Tip Displacement of the Cantilevered Beam as a function of A_1 and A_2

are shown in Figures 4 and 5. It can be clearly seen that the physics-based strategy works extremely well for this problem with maximum error of 0.35%.

We also applied the discrete formulation to a stepped cantilevered Euler-Bernoulli beam with two design variables. The discretized displacement vector for the case of 2 elements can be written as

$$\hat{\mathbf{w}} = \left\{ \hat{u}(x_1, A_1, A_2), \frac{\partial \hat{u}}{\partial x}(x_1, A_1, A_2), \hat{u}(x_2, A_1, A_2), \frac{\partial \hat{u}}{\partial x}(x_2, A_1, A_2) \right\}^T,$$

where \hat{u} is a feedforward neural network model with three inputs (i.e., x , A_1 , and A_2) which is employed as a surrogate model for the transverse displacement of the beam.

For this problem, we were able to drive the integral of the residual error to 5×10^{-3} using 20 nodes in

the hidden layer. The percentage error in the transverse displacement at the tip (i.e., x_2) were then computed over a range of values of A_1 and A_2 . The trends are shown in Figure 6. The results show that the mean and maximum percentage error are 0.78 and 2.12, respectively. This accuracy level is very impressive considering the fact that only 30 collocation points are chosen to compute the physics-based cost function.

8 Concluding Remarks

In this paper, we have introduced a physics-based unsupervised learning strategy for solving parameterized steady-state PDEs arising in the domain of design optimization and uncertainty analysis. In particular, we considered the cases when the parameters in the PDE operators are bounded design variables or random with specified density functions. Our approach interprets a parameterized PDE as a multidimensional operator problem with variable coefficients. It is shown that this interpretation allows us to postulate a model for the field variables in terms of the spatial coordinates and the parameters within the PDE operators.

We propose the application of neural networks and radial basis expansions which are capable of universal approximation to learn the underlying relationships. Continuum and discrete formulations of physics-based surrogate modeling have been presented for computing the undetermined parameters in the learning model. It is shown that both the formulations lead to a nonlinear programming problem. Some theoretical and implementation aspects of the formulations are also discussed. It is worth noting that our formulations unify both deterministic and stochastic system analysis since the probability measure used in the definition of the physics-based cost functions reduces to the Lebesgue measure for deterministic parameterization.

An important contribution of the present research is that - “*using universal learning models as parameter-dependent trial functions, a surrogate model can be constructed for deterministically and randomly parameterized PDEs without using any input-output training data.*” This has important ramifications for at least two application areas: firstly, design optimization of systems governed by steady-state PDEs, and secondly, numerical solution of PDEs whose coefficients are modeled as random or convex variables.

We presented some preliminary results for physics-based surrogate modeling of one-dimensional problems with two parameters. It is shown that the discrete formulation works extremely well for the prob-

lems considered. Further numerical studies are currently underway to test our formulations on more complex problems with larger number of variables.

We also outlined some avenues for extending the present approach to directly approximate the density function of the solution process. The notion of *weak* solutions and the principle of entropy maximization were leveraged to tackle the density estimation problem. Since the emphasis of the present work was on examining some of the theoretical issues relevant to this challenging problem, we have chosen to omit numerical studies. We hope that the ideas presented in this paper will lead to a more rigorous approach for modeling deterministic and uncertain systems governed by parameterized PDEs.

Acknowledgments

This work was supported by a grant from the School of Engineering Sciences at the University of Southampton. The author thanks Arindam Choudhury for useful discussions on density estimation.

References

¹Vapnik, V., *The Nature of Statistical Learning* New York: Springer Verlag, 1995.

²Bishop, C. M., *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford, 1995.

³MacKay, D. J. C., "Bayesian Methods for Adaptive Models," *PhD Thesis*, California Institute of Technology, 1992.

⁴Balabanov, V. O., Giunta, A. A., Golovidov, O., Grossman, B., Mason, W. H., Watson, L. T., and Haftka, R. T., "Reasonable Design Space Approach to Response Surface Approximation," *Journal of Aircraft*, Vol. 36, No. 1, 1999, pp. 308-315.

⁵Lin, Y., "Tensor Product Space ANOVA Models," *Annals of Statistics*, Vol. 28, No. 3, 2000, pp. 734-755.

⁶Kohonen, T., *Self-Organizing Maps*, Springer, 1995.

⁷Kansa, E. J., "Multiquadrics- A Scattered Data Approximation Scheme with Applications to Computational Fluid Dynamics: I. Surface Approximations and Partial Derivative Estimates," *Computers and Mathematics with Applications*, Vol. 19, No. 6-8, 1990, pp. 127-145.

⁸Kansa, E. J., "Multiquadrics- A Scattered Data Approximation Scheme with Applications to Computational Fluid Dynamics: II. Solutions to Parabolic, Hyperbolic, and Elliptic Partial Differential Equations," *Computers and Mathematics with Applications*, Vol. 19, No. 6-8, 1990, pp. 147-161.

⁹Lagaris, I. E., Likas, A. and Fotiadis, D. I., "Artificial Neural Networks for Solving Ordinary and Partial Differential Equations," *IEEE Transactions on Neural Networks*, Vol. 9, No. 5, 1998, pp. 987-1000.

¹⁰Lagaris, I. E., Likas, A. and Papageorgiou, D. G., "Neural Networks Methods for Boundary Value Prob-

lems with Irregular Boundaries," *IEEE Transactions on Neural Networks*, Vol. 11, No. 5, 2000, pp. 1041-1049.

¹¹Lagaris, I. E., Likas, A. and Fotiadis, D. I., "Artificial Neural Network Methods in Quantum Mechanics," *Computer Physics Communications*, Vol. 104, 1997, pp. 1-14.

¹²Belytschko, T., Krongauz, Y., Organ, D., Fleming, M., and Krysl, P., "Meshless Methods: An Overview and Recent Developments," *Computer Methods in Applied Mechanics and Engineering*, Vol. 139, 1996, pp. 3-47.

¹³Kirsch, U., "Reduced Basis Approximation of Structural Displacements for Optimal Design," *AIAA Journal*, Vol. 29, No. 10, 1991, pp. 1751-1758.

¹⁴Akgün, M. A., Garcelon, J. H., and Haftka, R. T., "Fast Exact Linear and Nonlinear Structural Reanalysis and the Sherman-Morrison-Woodbury Formulas," *International Journal of Numerical Methods in Engineering*, Vol. 50, No. 7, 2001, pp. 1587-1606.

¹⁵Nair, P. B., "Equivalence Between the Combined Approximations Technique and Krylov Subspace Methods," *AIAA Journal*, 2002, in press.

¹⁶Nair, P. B., Keane, A. J., and Langley, R. S., "Improved First-Order Approximation of Eigenvalues and Eigenvectors," *AIAA Journal*, Vol. 36, No. 9, 1998, pp. 1721-1727.

¹⁷LeGresley, P. A., and Alonso, J. J., "Airfoil Design Optimization Using Reduced Order Models Based on Proper Orthogonal Decomposition," *AIAA Paper 2000-2445*, June 2000.

¹⁸Nair, P. B., "On the Theoretical Foundations of Stochastic Reduced Basis Methods," *AIAA Paper 2001-1677*, April 2001.

¹⁹Nair, P. B., Choudhury, A., and Keane, A. J., "Some Greedy Algorithms for Sparse Nonlinear Regression," *Proceedings of the Eighteenth International Conference on Machine Learning*, Morgan Kaufmann, 2001, pp. 369-376.

²⁰Barron, R. A., "Universal Approximation Bounds for Superpositions of a Sigmoid Function," *IEEE Transactions on Information Theory*, Vol. 39, No. 3, 1993, pp. 930-945.

²¹Franke, C., and Schaback, R., "Solving Partial Differential Equations by Collocation Using Radial Basis Functions," *Applied Mathematics and Computation*, 93, 1998, pp. 73-83.

²²Jaynes, E. T., "On the Rationale of Maximum-Entropy Methods," *Proceedings of the IEEE*, Vol. 70. No. 9, 1982, pp. 939-952.

²³Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P., "Design and Analysis of Computer Experiments," *Statistical Science*, Vol. 4, No. 4, 1989, pp. 409-435.

²⁴B. F. Spencer and L. A. Bergman, "On the Numerical Solution of the Fokker-Planck Equation for Nonlinear Stochastic Systems," *Nonlinear Dynamics*, 4, 1992, pp. 357-372.