INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2010; **63**:121–162
Published online 8 July 2009 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.2112

Nonlinear system identification using radial basis functions

Paritosh Mokhasi and Dietmar Rempfer*, †

Department of Mechanical, Materials and Aerospace Engineering, Illinois Institute of Technology, Chicago, IL, U.S.A.

SUMMARY

Focussing on applications to fluid flow phenomena, we are interested in developing dynamical system models that are based on the discrete multivariate time series information only. A method based on radial basis functions and linear multi-step methods is used to construct continuous nonlinear models that approximate the original dynamical system. Information, such as the structure of the original system, is incorporated into the models through weak constraints. The formulation of the model and its advantages associated with modeling is described. Different examples are presented that highlight the various characteristics of the model and its effectiveness in dealing with various problems encountered in fluid flow problems. Copyright © 2009 John Wiley & Sons, Ltd.

Received 16 October 2008; Revised 12 May 2009; Accepted 14 May 2009

KEY WORDS: nonlinear dynamical systems; time series analysis; time series prediction; Navier–Stokes equations; turbulence; radial basis functions; proper orthogonal decomposition

1. INTRODUCTION

In many practical fields of engineering, there is an important need to construct dynamical systems from observed data. This is especially true in the field of fluid dynamics where the behavior of a fluid flow phenomenon can be understood through the use of dynamical systems. The governing equations associated with fluid flow—the Navier–Stokes equations—are intractable in their primitive form, which leads researchers to explore the dynamics of fluid flow either through experiments or through numerical simulations.

A popular approach to make the analysis of fluid flow problems more tractable is the method of proper orthogonal decomposition (POD). POD has long been used in the fluid dynamics community as a tool for the identification of coherent structures [1–5]. It has also been widely used in the field of data compression [6], signal processing applications, e.g. characterization of human faces [7, 8], and has been proven an ideal tool when dealing with large data sets. A number of properties

*Correspondence to: Dietmar Rempfer, Department of Mechanical, Materials and Aerospace Engineering, Illinois Institute of Technology, Chicago, IL, U.S.A.
†E-mail: rempfer@iit.edu

associated with POD and its application to fluids and complex dynamical systems have been studied in detail in the work of Aubry *et al.* [9, 10]. POD enables us to decompose a signal that varies in space and time into a set of basis functions that vary only in space, and a set of expansion coefficients that vary only in time. The appeal of this approach is that, since the spatial functions do not change in time, with known basis functions we can characterize the instantaneous state of the velocity field based on the expansion coefficients alone. Thus, we obtain approximations to the entire three-dimensional flow field that rely only on the information contained in an optimally small set of expansion coefficients. This small set of expansion coefficients requires significantly, often orders of magnitude less storage space than a straightforward collection of instantaneous velocity fields.

When the Navier–Stokes equations are projected onto the POD basis functions, we obtain evolution equations for the POD temporal coefficients using the Galerkin method [11]. The Galerkin model provides the evolution equations in the form of a quadratic system of ordinary differential equations (ODEs), which in principle can be solved numerically. However, in the Galerkin model, the appropriate modeling of the pressure terms and the POD truncation become important. The work of Rempfer [12, 13], Noack *et al.* [14–16], Sirisup and Karniadakis [17], and others highlight these problems and suggest means of overcoming them. Perret *et al.* [18] propose direct computation of the Galerkin coefficients using the least-squares method, while Galletti *et al.* [19] favor optimization methods to compute the correct coefficients. Similar methods have also been presented in the work of Couplet *et al.* [20]. However, these methods become computationally expensive when dealing with very high numbers of modes.

The problem of obtaining the evolution equations for the POD temporal coefficients can also be viewed from the perspective of parameter optimization. Since the Navier–Stokes equations explicitly dictate the form of the evolution equation, one can then look for methods to compute the coefficients in the system of differential equations. Parameter estimation and system identification have been studied and developed since the early 50s. The problem of parameter estimation can be described as follows. Given a system of ODEs as

$$\begin{aligned} \frac{\mathrm{d}\zeta_k}{\mathrm{d}t} &= F_k(\boldsymbol{\zeta},\theta), \quad k=1,2,\ldots,N, \quad t\in[0,T], \quad \boldsymbol{\zeta}(t)\in\mathbb{R}^d \\ \boldsymbol{\zeta}(t_0) &= \boldsymbol{\zeta}_0 \\ \eta_m(t_j) &= \zeta_m(t_j) \end{aligned} \tag{1}$$

where η_m are the observed discrete time series and θ represent the linear and quadratic coefficients that need to be computed. This leads to an optimization problem given by

$$\widehat{\theta}=\arg\min_{\theta}\left\{\sum_{j=1}^{P}\sum_{m=1}^{N}(\eta_m(t_j)-\zeta_m(t_j))^2\right\} \quad \text{subject to the constraint}$$

$$\frac{\mathrm{d}\zeta_k}{\mathrm{d}t}=F_k(\boldsymbol{\zeta},\theta), \quad \boldsymbol{\zeta}(t_0)=\boldsymbol{\zeta}_0$$

The above problem is generally solved using the nonlinear least-squares (NLS) approach. The papers of Bard [21] and Benson [22] describe examples of some of the earlier work in this area. Monte Carlo methods have also been used to tackle these problems and can be seen in the work of Carlin *et al.* [23] and Bremer and Kaplan [24], to name a few. Perhaps the main drawback of the NLS method is that it is computationally expensive and can only be used for very simple systems. There have been attempts to overcome these drawbacks through the method of contractive mapping [25]. Shooting methods have also been used as a popular and robust tool as demonstrated in the work by Muller and Timmer [26], Baake *et al.* [27], Bock [28] and Voss *et al.* [29], among others. Multiple shooting methods have been proven to be very robust and capable of handling partial data as well as noise. However, the method still remains computationally expensive. A different, more promising approach to parameter estimation comes in the form of basis function expansions for boundary value and distributed data problems. The use of spline bases for dynamic data fitting problems was demonstrated by Benson [22] and later by Varah [30]. Varah suggested to first fit the available data using data smoothing methods, and then use the time derivative approximations obtained from the fit to solve a least-squares problem, which maximizes the measure of fit between $\mathrm{d}\zeta_k/\mathrm{d}t$ and $F_k(\boldsymbol{\zeta}, \theta)$ with respect to the parameters θ. These methods were further refined by Ramsay *et al.* [31], and Poyton *et al.* [32] by using principle differential analysis as first described by Ramsay [33]. The method is again able to handle noisy and incomplete data and overcomes some of the computational drawbacks associated with the previous methods i.e. multiple shooting.

Our objective in this paper is to demonstrate a new non-iterative method based on radial basis functions and linear multi-step methods to generate a dynamical system from the multivariate time series of the POD expansion coefficients. This is done by describing the nonlinear term in the system of ODEs (in our case the quadratic term) with radial basis functions (RBF). We then weakly enforce the quadratic form of the differential equations through constraints. Such an approach allows us not only to obtain a nonlinear dynamical system, but also to approximate the linear and quadratic coefficients of the original dynamical system. For our case, the POD analysis provides us with a sample time series of the POD coefficients, which are then used to construct the dynamical system. The challenge associated with the proposed method lies in the fact that the dynamical system is often extremely high-dimensional whereas the sample time series is quite small. Furthermore, the data might be non-equally spaced and is of course discrete in nature.

RBF find their primary use in the field of scattered data interpolation and methods associated with them are often classified under the category of mesh-free methods in numerical analysis. The use of mesh-free methods for interpolation and approximation can be seen in the early works of Shepard [34], Hardy [35] and others. The main power of RBF comes from the fact that this method is able to handle high-dimensional data without incurring any additional computational cost [36], and it has been used in a variety of different fields. For example, RBF have been used in solving partial differential equations [37–40]. In the statistics literature, one can find a very close connection between kernel ridge regression [41–43] and RBF. Similarly, in the field of geo-statistics, one can find a similar connection to the method of Kriging [44] for interpolation. A detailed account of the use of RBF and their application is given in the book by Fasshauer [36]. RBF have also been called neural networks and have been introduced as such by Darken and Daley in 1989. The primary goal of such neural networks is to perform some form of nonlinear regression analysis. Examples of applications for prediction analysis in general can be seen in the work of Chng *et al.* [45], Casdagli [46, 47], Broomhead and Lowe [48], and Potts and Broomhead [49], to name a few. Applications also include the nonlinear prediction of chaotic time series as in the work of Chng *et al.* [45], Cowper *et al.* [50], Casdgali [47], Ni and Simons [51] and Mann and McLaughlin [52], among others. However, in most cases, the RBF networks are treated like a 'black box' and very little information, if any, about the original system can be inferred from the RBF networks.

Linear multi-step methods are a means of numerically integrating systems of ODEs. Details of the general linear multi-step method have long been established and we refer the reader to standard numerical analysis textbooks, such as Kincaid and Cheney [53] for details. The key idea of using multi-step methods is to avoid the approximation of the time derivative of the coefficients. We will instead use multi-step methods to directly approximate the *continuous* dynamical system.

The content of the paper is organized as follows. We first elaborate on the method of POD and the construction of the evolution equation associated with POD in Section 2. In Section 3, the basic equations associated with RBF are discussed. Sections 4 and 5 discuss the specifics associated with the development of the RBF evolution model and the various aspects associated with it. The method is then validated through the use of four examples in Section 6 followed by conclusions and future work in Section 7.

2. PROPER ORTHOGONAL DECOMPOSITION

The method of POD is a technique for decomposing a spatio-temporal signal into a set of orthogonal spatial basis functions and a set of temporal coefficients. This technique has found considerable success in the field of fluid mechanics for understanding turbulent flows and identification of coherent structures. With respect to our motivation, the method provides a means for fast computation of the evolution of fluid flow in a complex system. The basic decomposition of POD is given by

$$\mathbf{u}(\mathbf{x},t)=\sum_k \zeta_k(t)\boldsymbol{\psi}_k(\mathbf{x}) \tag{2}$$

The basis functions are computed via a minimization problem, which leads to an integral eigenvalue relation

$$\int_D R(\mathbf{x},\mathbf{x}')\boldsymbol{\psi}(\mathbf{x}')\,\mathrm{d}\mathbf{x}'=\vartheta\boldsymbol{\psi}(\mathbf{x}) \tag{3}$$

where

$$R(\mathbf{x},\mathbf{x}')=\frac{1}{T}\int \mathbf{u}(\mathbf{x},t)\mathbf{u}(\mathbf{x}',t)\,\mathrm{d}t \tag{4}$$

Thus, $R(\mathbf{x},\mathbf{x}')$ is the two-point auto correlation matrix, and the basis functions $\boldsymbol{\psi}(\mathbf{x})$ are the eigenvectors of the correlation matrix. However, when one has a domain with a large number of spatial points, the above approach for computing the basis functions becomes undesirable due to the computational constraints. Instead, we use the 'method of snapshots' as developed by Sirovich [5] to convert the spatial integral problem into a temporal problem. A correlation matrix is computed as

$$C_{ij}=\frac{1}{M}\int_D \mathbf{u}^{\mathrm{EN}}(\mathbf{x},t_i)\cdot\mathbf{u}^{\mathrm{EN}}(\mathbf{x},t_j)\,\mathrm{d}\mathbf{x} \tag{5}$$

where M is the number of snapshots available in the original ensemble of data and $\mathbf{u}^{\mathrm{EN}}$ represents the velocity snapshots that form the ensemble. Let v_i^k be the ith element of the eigenvector $\mathbf{v}^k$ corresponding to the eigenvalue ϑ_k, $k=1,2,3,\ldots,M$ of the correlation matrix $\mathbf{C}$. Let us define α_{ki} as

$$\alpha_{ki}:=\frac{v_i^k}{\sqrt{M\sum_{m,r}v_m^k v_r^k C_{mr}}} \tag{6}$$

Based on the above equations, one can then compute the POD basis functions as

$$\boldsymbol{\psi}_k(\mathbf{x}) = \sum_{i=1}^{M} \alpha_{ki} \mathbf{u}^{\mathrm{EN}}(\mathbf{x}, t_i) \tag{7}$$

and the temporal coefficients as

$$\zeta_k(t_p) = \int_D \mathbf{u}(\mathbf{x}, t_p) \boldsymbol{\psi}_k(\mathbf{x}) \, \mathrm{d}\mathbf{x} = M \sum_{i=1}^{M} \alpha_{ki} \widetilde{C}_{ip} \tag{8}$$

where

$$\widetilde{C}_{ip} = \int_D \mathbf{u}^{\mathrm{EN}}(\mathbf{x}, t_i) \mathbf{u}(\mathbf{x}, t_p) \, \mathrm{d}\mathbf{x} \tag{9}$$

One can also show that

$$\sum_{r=1}^{M} \zeta_k^{\mathrm{EN}}(t_r) \alpha_{mr} = \delta_{km} \tag{10}$$

The terms $\zeta_k^{\mathrm{EN}}(t_r) = \zeta_{kr}^{\mathrm{EN}}$ and $\mathbf{u}^{\mathrm{EN}}(\mathbf{x}, t_r) = \mathbf{u}_r^{\mathrm{EN}}$ are the POD coefficients and velocity fields within the ensemble, respectively. When all the M modes are considered, the matrix with elements ζ_{kr}^{EN} is the inverse of α_{rm}. We note that the POD coefficients within the ensemble ζ^{EN} can be easily computed by substituting $\widetilde{C}$ in (8) with C as given in (5). However, to compute POD coefficients outside the ensemble, one must perform the spatial integration of the velocity fields with the appropriate basis functions $\boldsymbol{\psi}(\mathbf{x})$, as given in (8). From the decomposition, the following property also holds:

$$\int_D \boldsymbol{\psi}_i(\mathbf{x}) \boldsymbol{\psi}_j(\mathbf{x}) \, \mathrm{d}\mathbf{x} = \delta_{ij} \tag{11}$$

$$\frac{1}{T} \int_0^T \zeta_i(t) \zeta_j(t) \, \mathrm{d}t = \langle \zeta_i(t) \zeta_j(t) \rangle = \vartheta_i \delta_{ij} \tag{12}$$

where ϑ_i are the eigenvalues associated with the eigenvectors $\mathbf{v}$. The properties (11) and (12) prompted the use of the term 'bi-orthogonal decomposition' in Aubry *et al.* [9]. The method of snapshots reduces the computational complexity significantly. The spectral convergence properties ensure that the basis functions are optimal and converge faster than any other set of functions that could be constructed in the Hilbert space. Furthermore, as seen in (7), all the basis functions are constructed as linear superpositions of the ensemble snapshots. Therefore, the basis functions automatically satisfy the continuity equation, no-slip and periodic boundary conditions (if such conditions apply). The conditions of mass conservation and no-slip at solid walls are of particular importance when dealing with flows around complex geometries, such as the ones typical for an urban setting.

At this point it becomes apparent that a good set of POD basis functions is one that captures a variety of flow phenomena with sufficient accuracy. In order to achieve this, a common practice is to run numerical simulations for the domain of interest and obtain sample snapshots of the 3D velocity fields that span a variety of flow states, or span a large interval of time. Generally, these 3D velocity field snapshots are obtained from numerical simulations with sufficient spatial and temporal resolution. When one is dealing with flow cases that exhibit diverse flow phenomena, it becomes important that the POD decomposition be able to capture all of them. To ensure this, the initial ensemble of velocity snapshots should contain velocity fields that correspond to different conditions. The general rule of thumb is to select snapshots that span the range of several integral time scales of the flow.

In order to construct an evolution equation for the POD coefficients $\zeta(t)$, the Galerkin approach may be used. In this method, the POD basis functions are projected onto the incompressible Navier–Stokes equations and integrated in space. The Navier–Stokes equations are given by

$$\frac{\partial \mathbf{u}}{\partial t} = -\frac{1}{\rho}\nabla p - (\mathbf{u}\cdot\nabla)\mathbf{u} + \nu\nabla^2\mathbf{u} \tag{13}$$

The continuity equation is not considered here because it is identically satisfied by the POD basis functions $\boldsymbol{\psi}_k(\mathbf{x})$. If the velocity expansion is given by

$$\mathbf{u}(\mathbf{x},t) = \mathbf{U}(\mathbf{x}) + \sum_{k=1}^{N} \zeta_k(t)\,\boldsymbol{\psi}_k(\mathbf{x}) \tag{14}$$

then projecting the Navier–Stokes equations onto the basis functions $\boldsymbol{\psi}_k(\mathbf{x})$ and integrating over the domain results in a set of ODEs of the form

$$\frac{\mathrm{d}\zeta_k}{\mathrm{d}t} = \sum_{i=1}^{N}\sum_{j=1}^{N} A_{kij}\zeta_i\zeta_j + \sum_{i=1}^{N} B_{ki}\zeta_i + C_k \tag{15}$$

where

$$A_{kij} = -\int_D \boldsymbol{\psi}_k\cdot(\boldsymbol{\psi}_i\cdot\nabla\boldsymbol{\psi}_j)\,\mathrm{d}\mathbf{x} + \mathscr{Q}_{kij} \tag{16}$$

$$B_{ki} = \nu\int_D \boldsymbol{\psi}_k\cdot\nabla^2\boldsymbol{\psi}_i\,\mathrm{d}\mathbf{x} - \int_D \boldsymbol{\psi}_k\cdot(\mathbf{U}\cdot\nabla\boldsymbol{\psi}_i)\,\mathrm{d}\mathbf{x} - \int_D \boldsymbol{\psi}_k\cdot(\boldsymbol{\psi}_i\cdot\nabla\mathbf{U})\,\mathrm{d}\mathbf{x} \tag{17}$$

$$C_k = \nu\int_D \boldsymbol{\psi}_k\cdot\nabla^2\mathbf{U}\,\mathrm{d}\mathbf{x} - \int_D \boldsymbol{\psi}_k\cdot(\mathbf{U}\cdot\nabla\mathbf{U})\,\mathrm{d}\mathbf{x} \tag{18}$$

The term A_{kij} comes from the convective term in the Navier–Stokes equation and the term $\mathscr{Q}$ corresponds to the pressure term. The term B_{ki} comes from the diffusion term. Now, Equation (15) is an autonomous system of equations that can be solved if the proper initial conditions are provided and if we have a complete set of basis functions. However, since the objective is to compute a low-dimensional model, only the low-order POD modes (which contain the majority of the energy) are used. This leads to a truncation of the set of POD basis functions, and the effect of truncation (or unresolved modes) must be modeled.

There exist a number of ways this could be achieved, some of which are discussed in [12, 14–17]. Furthermore, one must also find an appropriate means of modeling the pressure term. One method is to include the unresolved pressure term as part of the unresolved truncated modes and find a model that stabilizes the system and follows the dynamics correctly. The other approach is based on the fact that the pressure term is a nonlinear function of the velocity gradient as governed by the pressure-Poisson equation and therefore is modeled in the quadratic term. It should be noted that the second scenario is computationally expensive and not always used. We may therefore be interested in an alternative approach to compute the Galerkin coefficients.

One possible method to estimate the coefficients A_{kij} and B_{ki} in (15) is to obtain a system of linear equations for the unknown coefficients. If the system of ODEs consists of N equations, then there exists $(N+2)(N+1)/2$ number of unknowns for each differential equation. Therefore, in order to get reliable estimates of the coefficients one requires $P>(N+2)(N+1)/2$, where P is the number of POD coefficients available from the POD analysis, so that we have an over-determined system of equations, which can be solved using a least-squares approach. Such a constraint, therefore, only enables us to approximate dynamical systems of low dimensions. Furthermore, this method also requires time derivatives $\mathrm{d}\boldsymbol{\zeta}/\mathrm{d}t$ at the time steps t_p along with the coefficients $\boldsymbol{\zeta}(t_p)$. Other options include optimization methods and/or multiple shooting methods to estimate the coefficients. However, for high-dimensional systems, almost any optimization scheme is bound to be computationally expensive. Interestingly, Lorang *et al.* [54] showed that one does not require the such a large amount of data to compute the coefficients giving an indication that the quadratic term could in essence be represented by a fewer parameter terms than the conventional $(N)(N+1)/2$.

As an alternative means for overcoming some of the problems associated with the parameter estimation in (15), we abandon the conventional form assumed in (15) and use RBF to account for the nonlinear behavior of the system. Therefore, the right-hand side of (15) is now represented as

$$\frac{\mathrm{d}\zeta_k}{\mathrm{d}t}=\sum_{m=1}^{N}\mathscr{R}_{km}\zeta_m+\sum_{j=1}^{N}\lambda_{kj}\Phi(\|\mathbf{c}_j-\boldsymbol{\zeta}\|_2),\quad k=1,2,\ldots,N \tag{19}$$

where $\mathscr{R}$ and λ are coefficient matrices of the system that need to be computed. The form suggested in (19) is similar to a neural network, which had been used successfully in modeling nonlinear time series analysis and has been demonstrated as such in [54]. The RBF-evolution model consists of two parts, the linear part and the nonlinear part. The quadratic term has two major functions as pointed out by Noack *et al.* [55]. It acts like a energy sink on the high-frequency dynamics and for the low-frequency dynamics acts like a Reynolds stress component. It is therefore natural to find a substitute for the quadratic term in terms of a simpler nonlinear model. The various terms in (19) are as follows:

1. The term Φ is called a radial basic functions and is chosen *a-priori* from a family of positive definite functions.
2. The term $\mathbf{c}\in\mathbb{R}^{N\times 1}$ are called 'centers' and have the same dimension as the POD coefficients $\boldsymbol{\zeta}$. These centers are chosen to enforce certain properties, details of which are described in the following section.
3. $\mathscr{R}$ and λ are a set of coefficients that need to be determined.

By assuming the form in (19), we immediately eliminate one major issue, when the number of parameters to be computed may be higher than the number of data points available. The RBF model (19), in contrast, leads to a square system of equations that can be solved uniquely. Before going into the details of the RBF-evolution model, we will look at the basic equations and formulations associated with RBF.

3. RADIAL BASIS FUNCTIONS

In order to develop an evolution model based on RBF, we first consider the equations involved in the construction of the model. The RBF model always takes the form

$$g(\boldsymbol{\zeta})=\sum_j \lambda_j\Phi(\|\mathbf{c}_j-\boldsymbol{\zeta}\|) \tag{20}$$

where $g(\boldsymbol{\zeta})$ is an unknown function of $\boldsymbol{\zeta}=[\zeta_1,\zeta_2,\ldots,\zeta_n]$. However, the numerical values of g at different $\boldsymbol{\zeta}$'s are known. The λ_j are coefficients that need to be computed, $\mathbf{c}_j$ are called the centers and $\Phi(r)$ are the RBF, which are known. The RBF modeling procedure thus corresponds to a surface approximation in $\mathbb{R}^n$. For the purpose of RBF models, we have $\boldsymbol{\zeta}^j$ available at j different locations, i.e. $\boldsymbol{\zeta}^j=[\zeta_1(t_j),\zeta_2(t_j),\ldots,\zeta_n(t_j)]$, and we know the function values $g(\boldsymbol{\zeta}^j)$ at $\boldsymbol{\zeta}^j$. The radial functions are chosen positive definite. This property of the basis functions leads to a well-defined interpolation problem. Common examples of such functions are

$$\Phi_j(\mathbf{x})=\mathrm{e}^{(-\varepsilon^2\|\mathbf{x}-\mathbf{c}_j\|_2^2)} \quad \text{Gaussian function}$$

$$\Phi_j(\mathbf{x})=\sqrt{1+\varepsilon^2\|\mathbf{x}-\mathbf{c}_j\|_2^2} \quad \text{multi-quadric function}$$

$$\Phi_j(\mathbf{x})=\frac{1}{\sqrt{1+\varepsilon^2\|\mathbf{x}-\mathbf{c}_j\|_2^2}} \quad \text{inverse multi-quadric function}$$

The term ε in the functions is called the scaling (or shape) parameter that governs how narrow or wide the radial function should be. The term $\|\mathbf{x}-\mathbf{c}_j\|$ is the Euclidian distance between the centers $\mathbf{c}_j$ and the data points $\mathbf{x}$.

The interpolation problem is now reduced to computing the coefficients λ_j and the shape parameter ε. Identification of the coefficients and shape parameter would ideally lead to a nonlinear optimization problem. However, it is general practice to assume a constant shape parameter for all the basis functions, which then leads to a system of linear equations given by

$$g(\boldsymbol{\zeta}^i)=\sum_j \lambda_j\Phi(\|\mathbf{c}_j-\boldsymbol{\zeta}^i\|)=\Phi_{ij}\lambda_j \tag{21}$$

By choosing the centers $\mathbf{c}_j$ to be equal to the data locations $\boldsymbol{\zeta}^j$, the resulting matrix is a symmetric positive-definite matrix, which has an inverse. It should also be noted that the positive definiteness of the matrix is not affected by the value of the shape parameter ε. Therefore, the interpolation problem is naturally well defined, and the coefficients λ_j are computed as

$$\lambda_j=\Phi_{ij}^{-1}g(\boldsymbol{\zeta}^i) \tag{22}$$

This completes the RBF modeling procedure. The optimal shape parameter ε is computed using a cross-validation procedure and is generally problem dependent. We discuss the cross validation procedure in detail in the sections below.

4. RBF EVOLUTION MODEL

We now discuss two RBF-based formulations for constructing evolution models for the POD coefficients. We may envision two possible scenarios. In the first scenario, we are given the POD coefficients $\boldsymbol{\zeta}(t_p)$ and their time derivatives $\mathrm{d}\boldsymbol{\zeta}/\mathrm{d}t$. In the second scenario, we are given pairs of data points i.e. we are given $\boldsymbol{\zeta}(t_p)$ and $\boldsymbol{\zeta}(t_p+\delta t)$, where $p=1,2,\ldots,P$. Of course, one could argue that the second scenario could be converted to the first scenario by simply approximating the time derivative using the pairs of points. However, by approximating the time derivative as

$$\frac{\mathrm{d}\boldsymbol{\zeta}}{\mathrm{d}t}\approx\frac{\boldsymbol{\zeta}(t+\delta t)-\boldsymbol{\zeta}(t)}{\delta t}$$

one may introduce an approximation error into the model that cannot be easily removed. During the RBF modeling procedure, the error due to the approximations, therefore, may create a sub-optimal model whose behavior is not indicative of the true dynamical system. In the following subsections, we describe a method for the construction of the evolution models for the two scenarios.

4.1. Evolution model when time derivatives are given

We first address the case when the POD coefficients $\boldsymbol{\zeta}(t_p)$ and their time derivative $\mathrm{d}\boldsymbol{\zeta}/\mathrm{d}t$ are given. For this case, we are looking to construct the original dynamical system (15). The RBF approximation to the time derivatives is written as

$$F_k \equiv \frac{\mathrm{d}\zeta_k}{\mathrm{d}t} = \sum_{m=1}^{N} \mathscr{R}_{km}\zeta_m + \sum_{j=1}^{M} \lambda_{kj}\Phi(\|\mathbf{c}_j - \boldsymbol{\zeta}\|_2), \quad k = 1, 2, \ldots, N \tag{23}$$

The first step is to choose the appropriate centers $\mathbf{c}_j$ for the model. RBF theory recommends that using the centers equal to the data sites leads to a well-posed problem. Therefore, we set the centers $\mathbf{c}_j$ equal to the POD coefficients obtained from the POD analysis. Thus, $\mathbf{c}_j = \boldsymbol{\zeta}^{\mathrm{EN}}(t_j)$, $j = 1, 2, \ldots, P$ where P is the number of POD coefficients available.

Now, within this scenario, one could approach the RBF modeling problem in two ways. The first approach is to apply the standard RBF method to construct a system of linear equations for computing the coefficients $\mathscr{R}$ and λ. Note that for each k in (23), one has $P+N$ number of unknowns. An additional set of N equations is required to make the system unique. This is done by enforcing additional constraints given by

$$\sum_{j=1}^{j=P} \zeta_k^{\mathrm{EN}}(t_j)\lambda_{kj} = 0, \quad k = 1, 2, \ldots, N \tag{24}$$

Such a constraint is considered *natural* when one looks to model a linear function. The main objective of the above N equations is to provide a set of linearly independent functions. Furthermore, from a polynomial interpolation perspective, the conditions (24) guarantee polynomial precision if the underlying function F_k in (23) is linear in nature. Of course, it should be recognized that the choice of the additional N equations is somewhat arbitrary and one could, in principle, use any other set of independent functions. This idea is explored in the later part of this section. Now, based on the constraints (24), the RBF modeling problem can be written in matrix-vector form as

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{O} & \mathbf{C} \end{bmatrix} \begin{bmatrix} \mathscr{R} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{H} \\ \mathbf{O} \end{bmatrix} \tag{25}$$

where the elements of the sub-matrices are as follows:

$$A_{pm} = \zeta_m(t_p), \quad m = 1, 2, \ldots, N, \quad p = 1, 2, \ldots, P \tag{26}$$

$$B_{pj} = \Phi(\|\boldsymbol{\zeta}^{\mathrm{EN}}(t_j) - \boldsymbol{\zeta}^{\mathrm{EN}}(t_p)\|_2), \quad j = 1, 2, \ldots, P, \quad p = 1, 2, \ldots, P \tag{27}$$

$$C_{ml}=\zeta_m^{\mathrm{EN}}(t_l), \quad m=1,2,\ldots,N, \;\; l=1,2,\ldots,P \tag{28}$$

$$O_{mn}=0, \quad m=1,2,\ldots,N, \;\; n=1,2,\ldots,N \tag{29}$$

$$H_{pk}=\left(\frac{\mathrm{d}\zeta_k}{\mathrm{d}t}\right)_{t=t_p}, \quad p=1,2,\ldots,P, \;\; k=1,2,\ldots,N \tag{30}$$

We first note that this method produce a perfectly valid continuous RBF evolution model. However, when one knows the form of the original dynamical system that is being modeled, then one can incorporate that information into the RBF modeling procedure thereby producing better results. In our case, we are interested in using RBF to model systems of ODEs that have a quadratic form. In order to incorporate this information, we return to (15). We now rewrite the system of ODEs (15) into a system of *partial differential equations*. To do this, we differentiate (15) with respect to ζ_m, which gives

$$\frac{\partial F_k}{\partial \zeta_m}=B_{km}+\sum_{r=1}^{N}(A_{krm}+A_{kmr})\zeta_r$$

$$\left(\frac{\partial F_k}{\partial \zeta_m}\right)_{\boldsymbol{\zeta}=0}=B_{km}$$

Differentiating again with respect to ζ_l gives

$$\left(\frac{\partial^2 F_k}{\partial \zeta_m \partial \zeta_l}\right)_{\boldsymbol{\zeta}=0}=(A_{klm}+A_{kml})$$

We now write the Taylor series of $F_k(\boldsymbol{\zeta})$ about the origin, i.e. about $\boldsymbol{\zeta}=0$ which gives

$$F_k^p=\cancelto{0}{F_k(\boldsymbol{\zeta}=0)}+\sum_{n=1}^{N}\left[\zeta_n^p\left(\frac{\partial F_k}{\partial \zeta_n}\right)_{\boldsymbol{\zeta}=0}+\frac{1}{2}(\zeta_n^p)^2\left(\frac{\partial^2 F_k}{\partial \zeta_n^2}\right)_{\boldsymbol{\zeta}=0}\right]+\sum_{i=1}^{N}\sum_{j=i+1}^{N}(\zeta_i^p\zeta_j^p)\left(\frac{\partial^2 F_k}{\partial \zeta_i \partial \zeta_j}\right)_{\boldsymbol{\zeta}=0} \tag{31}$$

where $F_k^p=F_k(\boldsymbol{\zeta}^p)$, $\boldsymbol{\zeta}^p=\boldsymbol{\zeta}(t_p)$ and $F_k(\boldsymbol{\zeta}=0)=0$. Substituting (23) into the right-hand side of (31) gives

$$F_k^p=\sum_{r=1}^{N}\mathscr{R}_{kr}\zeta_r^p+\sum_{l=1}^{M}\lambda_{kl}\mathscr{Q}_l^p \tag{32}$$

where

$$\mathscr{Q}_l^p=\Phi(\|\boldsymbol{\zeta}_l^{\mathrm{EN}}\|)+\sum_{k=1}^{N}\left[\zeta_k^p\left(\frac{\partial \Phi_l}{\partial \zeta_k}\right)_{\boldsymbol{\zeta}^n=0}+\frac{1}{2}(\zeta_k^p)^2\left(\frac{\partial^2 \Phi_l}{\partial \zeta_k^2}\right)_{\boldsymbol{\zeta}^n=0}\right]$$
$$+\sum_{i=1}^{N}\sum_{j=i+1}^{N}(\zeta_i^p\zeta_j^p)\left(\frac{\partial^2 \Phi_l}{\partial \zeta_i \partial \zeta_j}\right)_{\boldsymbol{\zeta}^n=0} \tag{33}$$

with $\Phi_l = \Phi(\|\zeta_l^{\mathrm{EN}} - \zeta\|_2)$, where $\zeta_l^{\mathrm{EN}} := \zeta^{\mathrm{EN}}(t_l)$. Therefore, the sub-matrix $\mathbf{B}$ in (25) is now replaced with $\mathscr{Q}$, i.e. $\mathscr{B}_{pj} = \mathscr{Q}_j^p$. Once the coefficients $\mathscr{R}$ and λ are computed from the system of linear equations (25), the coefficients associated with the quadratic system of ODEs (15) can be calculated as

$$\begin{aligned} B_{kj} &= \sum_l \lambda_{kl} \left(\mathscr{R}_{lj} + \left(\frac{\partial \Phi_l}{\partial \zeta_j} \right)_{\zeta^n = 0} \right) \\ A_{kij} &= \sum_l \lambda_{kl} \left(\frac{\partial^2 \Phi_l}{\partial \zeta_i \partial \zeta_j} \right)_{\zeta^n = 0}, \quad i = 1, 2, \ldots, N, \quad j = i, \; i+1, \ldots, N \end{aligned} \tag{34}$$

Once the coefficients are computed, one can use the original quadratic form of the dynamical simulation for performing numerical integration. At this point it is important to note that if one wishes to use the RBF form of the dynamical system for performing integration, then one must use the form given in (32) rather than (23). This is because the coefficients in the RBF system are constructed using the form in (33). It is also important to note that this procedure also gives us an algorithm for parameter estimation using RBF.

4.2. Evolution model when data pairs are given

In this subsection we outline the procedure for constructing a RBF evolution model when pairs of POD coefficients, rather than time derivatives, are known. In other words, we are given $\zeta(t_p)$, $\zeta(t_p + \delta t)$, $p = 1, 2, \ldots, P$. As mentioned earlier, rather than trying to find a first-order time derivative from the data pairs, we construct a discrete RBF evolution map and then from this map construct a continuous dynamical model using linear multi-step methods.

The equations associated with the construction of the discrete RBF map remain largely unchanged from the previous cases (when time derivatives are known). The RBF discrete map takes the form

$$\zeta_k(t + \delta t) = \sum_{j=1}^{N} \mathscr{M}_{kj} \zeta_j + \sum_{m=1}^{M} \beta_{km} \Phi(\|\mathbf{c}_m - \zeta(t)\|_2), \quad k = 1, 2, \ldots, N \tag{35}$$

We now wish to compute the terms $\mathscr{M}$ and β in the above equation. As before, the centers $\mathbf{c}_j$ are chosen to be the available data sites and therefore are $\mathbf{c}_m = \zeta^{\mathrm{EN}}(t_m)$. In matrix-vector form, the system of equations to be solved is then

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{O} & \mathbf{C} \end{bmatrix} \begin{bmatrix} \mathscr{M} \\ \beta \end{bmatrix} = \begin{bmatrix} \mathbf{H} \\ \mathbf{O} \end{bmatrix} \tag{36}$$

where the elements of the various sub-matrices are

$$A_{pm} = \zeta_m(t_p), \quad m = 1, 2, \ldots, N, \quad p = 1, 2, \ldots, P \tag{37}$$

$$B_{pj} = \Phi(\|\zeta^{\mathrm{EN}}(t_j) - \zeta^{\mathrm{EN}}(t_p)\|_2), \quad j = 1, 2, \ldots, P, \quad p = 1, 2, \ldots, P \tag{38}$$

$$C_{ml} = \zeta_m^{\mathrm{EN}}(t_l), \quad m = 1, 2, \ldots, N, \quad l = 1, 2, \ldots, P \tag{39}$$

$$O_{mn} = 0, \quad m = 1, 2, \ldots, N, \quad n = 1, 2, \ldots, N \tag{40}$$

$$H_{pk} = \zeta_k(t_p + \delta t), \quad p = 1, 2, \ldots, P, \quad k = 1, 2, \ldots, N \tag{41}$$

We now address the question of constructing a continuous dynamical system using a discrete RBF evolution map. To achieve this objective, we make use of linear multi-step methods. Linear multi-step methods are means of numerically integrating systems of ODEs. The basic formulation of a linear multi-step method relies on the principle that the system of ODEs can be solved by integrating the time derivatives to give

$$\int_{t_p}^{t_p+\delta t} \frac{\mathrm{d}\zeta_k}{\mathrm{d}t}\,\mathrm{d}t = \zeta_k(t_p+\delta t)-\zeta_k(t_p) \tag{42}$$

and then

$$\zeta_k(t_p+\delta t)=\zeta_k(t_p)+\int_{t_p}^{t_p+\delta t} F_k(\boldsymbol{\zeta}(\tau))\,\mathrm{d}\tau \tag{43}$$

The integral on the right-hand side above is approximated using a numerical quadrature scheme, which results in a general expression given by

$$\zeta_k(t_p+\delta t)=\zeta_k(t_p)+\sum_{s=a}^{s=b} W_s F_k(\boldsymbol{\zeta}(t_p+s\cdot\delta t)) \tag{44}$$

Depending on the index values a and b, one can form either implicit or explicit multi-step methods. The integration weights W_s may be computed from the requirement that the approximation to the integrand

$$\int_{t_p}^{t_p+\delta t} F_k(\boldsymbol{\zeta}(\tau))\,\mathrm{d}\tau \approx \sum_{s=a}^{s=b} W_s F_k(\boldsymbol{\zeta}(t_p+s\cdot\delta t)) \tag{45}$$

be exact whenever the integrand is a polynomial of a certain order. By setting $a=-3$ and $b=1$ and requiring that the integrand is approximated by a fifth-order polynomial leads to a popular implicit formula known as the Adams–Moulton formula and is given by

$$\zeta_k(t_p+\delta t)=\zeta_k(t_p)+\sum_{s=-3}^{s=1} W_s F_k(\boldsymbol{\zeta}(t_p+s\cdot\delta t)),\quad k=1,2,\ldots,N \tag{46}$$

where

$$W_1=\frac{251\delta t}{720},\quad W_0=\frac{646\delta t}{720},\quad W_{-1}=-\frac{246\delta t}{720},\quad W_{-2}=\frac{106\delta t}{720},\quad W_{-3}=-\frac{19\delta t}{720} \tag{47}$$

For the purpose of this paper, the Adams–Moulton (A-M) formula is used to construct the continuous dynamical system. This formula is chosen because one can show that the method satisfies the conditions of stability and consistency. The (A-M) formula can be formally shown to be strongly stable, and the higher order A-M scheme, as the one used for this paper, could to use for stiff equations. It should be noted here that the choice of the multi-step method is not critical as long as it satisfies the stability and consistency conditions. The use of Runge–Kutta schemes is not considered feasible here because the use of the RK schemes would result in a system of nonlinear equations, which must be solved. Multi-step methods on the other hand does not have such a constraint.

Note that for the case considered here, the POD coefficients are available only as pairs of coefficients. In order to compute the continuous dynamical system using the Adams–Moulton formula, one requires the POD coefficients at a few extra time steps as well, namely at time $t_p-\delta t$, $t_p-2\delta t$, $t_p-3\delta t$ which are unknown for our case. However, one can compute the POD coefficients at the past time steps using the discrete RBF map. The procedure is:

1. Using (35), recursively find the POD coefficients $\boldsymbol{\zeta}(t_p+2\delta t)$, $\boldsymbol{\zeta}(t_p+3\delta t)$, $\boldsymbol{\zeta}(t_p+4\delta t)$, $p=1,2,\ldots,P$, where P is the number of data pairs available.
2. The centers $\mathbf{c}_j$ are set equal to $\boldsymbol{\zeta}(t_j)$, where $j=1,2,\ldots,P$.
3. The RBF evolution form is now chosen as given in (23).
4. The RBF form of F_k is now substituted into (46). This results in a system of equations that can be solved for the unknown RBF coefficients. The resulting RBF model will now be a continuous dynamical model.

Substitution of the RBF form (23) in (46) gives

$$\zeta_k(t_p+\delta t)-\zeta_k(t_p)=\sum_{m=1}^{N}\widehat{\mathscr{R}}_{km}\left(\sum_{r=-3}^{r=1}W_r\,\zeta_m(t_p+r\cdot\delta t)\right)+\sum_{j=1}^{M}\widehat{\lambda}_{kj}\left(\sum_{r=-3}^{r=1}W_r\,\Phi(\|\mathbf{c}_j-\boldsymbol{\zeta}(t_p+r\cdot\delta t)\|_2)\right) \tag{48}$$

where

$$W_1=\frac{251\delta t}{720},\quad W_0=\frac{646\delta t}{720},\quad W_{-1}=-\frac{246\delta t}{720},\quad W_{-2}=\frac{106\,\delta t}{720},\quad W_{-3}=-\frac{19\delta t}{720}$$

To complete the RBF modeling procedure, we require an additional N equations. As outlined in the earlier procedure, a possible set of N linear equations can be written as

$$\sum_{j=1}^{j=P}\zeta_k^{\mathrm{EN}}(t_j)\widehat{\lambda}_{kj}=0,\quad k=1,2,\ldots,N \tag{49}$$

These constraints can be considered a natural choice if the surface being approximated has a linear form. However, if one intends to enforce a constraint that the model should imitate a quadratic dynamical system, then one can do so by enforcing it as a weak constraint, that can be obtained by multiplying both sides of (32) with $\zeta_m(t_p)$ and integrating/summing over time t_p, $p=1,2,\ldots,P$

$$0=\sum_{j=1}^{j=P}\widehat{\lambda}_{kj}\underbrace{\langle\mathscr{Q}_j^p-\Phi(\|\mathbf{c}_j-\boldsymbol{\zeta}(t_p)\|_2),\zeta_m(t_p)\rangle}_{\mathbf{C}_{jm}},\quad m=1,2,\ldots,N \tag{50}$$

where $\langle .,.\rangle$ denotes integration over time and $\mathscr{Q}_j^p$ follows from (33). It is important to note that (50) is identical to

$$\left\langle\zeta_k,\frac{\mathrm{d}\zeta_k}{\mathrm{d}t}\right\rangle=\sum_{i=1}^{N}\sum_{j=1}^{N}A_{kij}\langle\zeta_k,\zeta_i\zeta_j\rangle+\sum_{l=1}^{N}B_{kl}\langle\zeta_k,\zeta_l\rangle \tag{51}$$

The left-hand side of the above equation represents the rate of change of kinetic energy. The change in kinetic energy is due to diadic and triadic interactions between the different POD coefficients. The details of such interactions have been elaborated in [12, 16, 55] to name a few. It is interesting to note that a form such as (51) is used to construct eddy-viscosity models, as done in [12] so as to stabilize the original dynamical system.

Now using (48) and (50) gives us P equations for P unknowns. Written in matrix-vector form we have

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{O} & \mathbf{C} \end{bmatrix} \begin{bmatrix} \widehat{\mathscr{R}} \\ \widehat{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{H} \\ \mathbf{O} \end{bmatrix} \tag{52}$$

where the elements of the sub-matrices are

$$A_{pm} = \sum_{r=-3}^{r=1} W_r \zeta_m(t_p + r \cdot \delta t), \quad m = 1, 2, \ldots, N, \ \ p = 1, 2, \ldots, P \tag{53}$$

$$B_{pj} = \sum_{r=-3}^{r=1} W_r \Phi(\|\mathbf{c}_j - \boldsymbol{\zeta}(t_p + r \cdot \delta t)\|_2), \quad j = 1, 2, \ldots, P, \ \ p = 1, 2, \ldots, P \tag{54}$$

$$C_{ml} = \sum_{p=1}^{P} \zeta_m(t_p)(\mathscr{Q}_l^p - \Phi(\|\mathbf{c}_l - \boldsymbol{\zeta}(t_p)\|_2)), \quad m = 1, 2, \ldots, N, \ \ l = 1, 2, \ldots, P \tag{55}$$

$$O_{mn} = 0, \quad m = 1, 2, \ldots, N, \ \ n = 1, 2, \ldots, N \tag{56}$$

$$H_{pk} = \zeta_k(t_p + \delta t) - \zeta_k(t_p), \quad p = 1, 2, \ldots, P, \ \ k = 1, 2, \ldots, N \tag{57}$$

5. SELECTION OF SCALING PARAMETER

As mentioned in the previous section, in order to uniquely define the problem of approximation using RBF, one must specify two important parameters. The first parameter is the location of the centers. This problem is addressed by making the location of the centers equal to the location of the data sites. The second parameter is the scaling parameter and it is intricately connected to the accuracy of the approximation. The scaling parameter ε is a scalar term that determines the width of the basic function. The underlying principle of the scaling parameter selection is to use a leave-one-out-cross validation (LOOCV) procedure. In the statistics literature, this method is also called predictive sum of squares (PRESS). Rippa [56] formulated the LOOCV procedure specifically for RBF and we use the equations developed in [56] for our case.

As seen in the previous section, we considered two different cases for which RBF evolution models could be generated. The first case assumed that the time derivatives of the POD coefficients along with the coefficients were available. The second case assumed that pairs of POD coefficients, separated by time increment δt, were known. Though both methods use the LOOCV criteria to compute the 'optimal' scaling parameter, the cost function to be minimized is different for the two cases. We now elaborate on the cost functions to be minimized for the two cases.

5.1. Cost function for when time derivatives are given

The cost function for this case directly reflects on the way, the RBF models are constructed. As discussed in the previous section, for the case when the time derivatives are given we form a RBF model by approximating the system of ODEs as a system of PDEs using a Taylor series expansion.

Therefore, we look for a cost function that reflects the property that the RBF is trying to model a partial differential equation. In order to find the optimal scaling parameter for this case, we first introduce the method for computing these parameters when one wishes to solve partial differential equations using RBF. This method has been elaborated on in detail in [57]. We first discuss the main points, in general, and then apply it to our case at the end.

Consider a solution f to a differential equation with a differential operator $\mathscr{L}$ given by

$$f(\mathbf{x}) = \sum_{j=1}^{N} \lambda_j \Phi_j(\|\mathbf{a}_j - \mathbf{x}\|_2), \quad \mathbf{x} \in \Omega \subset \mathbb{R}^s \tag{58}$$

where f is a scalar and $\mathbf{a}$ are the centers. Applying the differential operator on the above expansion gives

$$\mathscr{L} f(\mathbf{x}) = \sum_{j=1}^{N} \lambda_j \mathscr{L} \Phi_j(\|\mathbf{a}_j - \mathbf{x}\|_2) \tag{59}$$

If we now assume that the differential equation must be satisfied exactly at the collocation points $\mathbf{x}_i$, then we have

$$\mathscr{L} f(\mathbf{x}_i) = \sum_{j=1}^{N} \lambda_j \mathscr{L} \Phi_j(\|\mathbf{a}_j - \mathbf{x}_i\|_2) \tag{60}$$

Written in matrix-vector form, the above equations become

$$\begin{aligned} \mathscr{L} f &= \mathbf{A}_{\mathscr{L}} \lambda \\ &= \underbrace{\mathbf{A}_{\mathscr{L}} \mathbf{A}^{-1}}_{\mathbf{D}} f \end{aligned} \tag{61}$$

where $\mathbf{A}_{ij} = \Phi_j(\mathbf{x}_i) = \Phi(\|\mathbf{a}_j - \mathbf{x}_i\|, \varepsilon)$ and $\mathbf{A}_{\mathscr{L}} = \mathscr{L}\Phi_j(\|\mathbf{a}_j - \mathbf{x}_i\|_2, \varepsilon)$. The term $\mathbf{D}$ can be viewed as the discretized version of the differential operator. It is important to note that the differential operator $\mathscr{L}$ operates only on $\mathbf{x}$. Now

$$\mathbf{D} = \mathbf{A}_{\mathscr{L}} \mathbf{A}^{-1} \tag{62}$$

$$\mathbf{A}\mathbf{D}^{\mathrm{T}} = \mathbf{A}_{\mathscr{L}}^{\mathrm{T}} \tag{63}$$

where the superscript T denotes the matrix transpose. The objective now is to find the optimal scaling parameter ε such that (63) is as accurate as possible. The cost function that achieves this is given as

$$C(\varepsilon) = \left\| \frac{\sum_{j=1} \sum_{m=1} \mathbf{A}_{km}^{-1} (\mathbf{A}_{\mathscr{L}}^{\mathrm{T}})_{mj}}{A_{kk}^{-1}} \right\| \tag{64}$$

This ends construction of the cost function required that needs to be minimized in order to find the optimal scaling parameter ε when one wishes to solve a differential equation using an RBF expansion as given in (58).

In order to make the cost function suitable for our purposes, we need to find the appropriate differential operator. To do this, we first recall that the quadratic system of ODEs in (31) can be written in a partial differential form as

$$0 = F_k(\boldsymbol{\zeta}) + \sum_{k=1}^{N}\left[\frac{1}{2}(\zeta_k)^2\frac{\partial^2 F_k}{\partial \zeta_k^2} - \zeta_k\frac{\partial F_k}{\partial \zeta_k}\right] + \sum_{i=1}^{N}\sum_{j=i+1}^{N}(\zeta_i\zeta_j)\frac{\partial^2 F_k}{\partial\zeta_i\partial\zeta_j} \tag{65}$$

Therefore, the appropriate differential operator for our case is found to be

$$\mathscr{L} = 1 + \sum_{k=1}^{N}\left[\frac{1}{2}(\zeta_k)^2\frac{\partial^2}{\partial \zeta_k^2} - \zeta_k\frac{\partial}{\partial \zeta_k}\right] + \sum_{i=1}^{N}\sum_{j=i+1}^{N}(\zeta_i\zeta_j)\frac{\partial^2}{\partial\zeta_i\partial\zeta_j} \tag{66}$$

and the term $\mathbf{A}_{\mathscr{L}}$ in (64) has elements

$$(\mathbf{A}_{\mathscr{L}})_{ij} = \mathscr{L}\Phi(\|c_j - \boldsymbol{\zeta}\|)|_{\boldsymbol{\zeta}=\boldsymbol{\zeta}_i} \tag{67}$$

It should be noted that this procedure also results in a single scaling parameter ε for the entire system of differential equations. This method, in some sense, weakly enforces the condition that the underlying dynamical system that is being modeled is quadratic in nature.

5.2. *Cost function when pairs of coefficients are given*

To formulate the cost function for this case, we again turn to the method in which the RBF procedure was used to model the dynamical system. Referring back to the previous section associated with the case when pairs of coefficients are given, we see that the RBF model now reflects more toward a standard multivariate surface approximation problem. In this case, the surface being approximated is the manifold that lies on the N-dimensional state space. For the standard interpolation problem of

$$g(\mathbf{x}) = \sum_{j=1}^{N}\lambda_j\,\Phi_j(\mathbf{x})$$

the cost function is given by

$$C(\varepsilon) = \left\|\frac{\widehat{A}_{km}^{-1}(\varepsilon)\,g_m}{\widehat{A}_{kk}^{-1}(\varepsilon)}\right\| \tag{68}$$

where

$$\widehat{A}_{km} = \Phi(\|\mathbf{c}_k - \boldsymbol{\zeta}_m\|, \varepsilon)$$

and $g_m = g(\mathbf{x}_m)$. However, for our case, as seen in (36) and (52), we have an additional linear term in the RBF approximation, which must also be taken into account. Specifically, the system of equations (in matrix-vector form) is

$$\begin{bmatrix}\mathbf{A} & \mathbf{B}(\varepsilon)\\ \mathbf{O} & \mathbf{C}\end{bmatrix}\begin{bmatrix}\widehat{\mathscr{R}}\\ \widehat{\lambda}\end{bmatrix} = \begin{bmatrix}\mathbf{g}\\ \mathbf{O}\end{bmatrix}$$

Therefore, the modified cost function for the above case is given by

$$C(\varepsilon) = \left\| \frac{\sum_m \sum_l B_{km}^{-1}(\varepsilon)\, G_{ml}(\varepsilon)\, g_l}{B_{kk}^{-1}(\varepsilon)} \right\| \tag{69}$$

where

$$\mathbf{G}(\varepsilon) = \mathbf{I} - \mathbf{A}(\mathbf{C}\mathbf{B}^{-1}(\varepsilon)\,\mathbf{A})^{-1}(\mathbf{C}\mathbf{B}^{-1}(\varepsilon)) \tag{70}$$

where $\mathbf{I}$ is an identity matrix with same size as $\mathbf{B}$. The matrix $\mathbf{G}$ is obtained by eliminating the set of equations associated with $\mathbf{C}$ and computing $\widehat{\mathscr{R}}$ in terms of $\mathbf{g}$ only.

6. RESULTS AND DISCUSSION

We now present four different examples to demonstrate the effectiveness of the RBF models. The 3D Lorenz model, the 9D Lorenz model, a 5D nonlinear model and the Kuramoto–Sivashinsky (KS) equation are considered. The first two examples are used to demonstrate the use of RBF evolution models for quadratic systems of ODEs. The third example is used to demonstrate the flexibility of our RBF approach to model general nonlinear dynamical systems. As a final example, we use a spatio-temporal system to demonstrate the use of the RBF models to estimate the evolution of a spatio-temporal phenomenon.

As mentioned in the previous section, we explore the use of four different types of RBF evolution models depending on the type of input information available to us. If pairs of solution snapshots are given at discrete time steps, then we construct a discrete RBF evolution map and based on it construct an approximation to the original dynamical system. If time derivatives of the solution along with the snapshots themselves are known at discrete locations, then two different RBF evolution models are constructed. These models are continuous in nature. The first model, in principle, can be used to describe any nonlinear dynamical system while the second model focuses on the case when the original dynamical system has a quadratic polynomial form.

Before looking at the RBF models, we first describe the various examples that are used in this section. To facilitate the explanation of the RBF evolution models, we define a ‘snapshot’ as the solution of the dynamical system at a certain instance in time t. A single snapshot at time t consists of two separate pieces of information, namely, the solution at time t *and* the solution at time $t+\delta t$ **or** the time derivative at time t (depending on the case we are examining). A ‘database’ is referred to as a collection of snapshots that is used to construct the RBF evolution model. We further classify the database into ‘database A’ and ‘database B’. The snapshots in database A consist of the solution at time t_n and at time $t_n+\delta t$, $n=1,2,\ldots,N$. The snapshots in database B consist of the solution and its time derivative. The snapshots within the database are generally taken randomly from within a time range. The specifics of the number of snapshots within a database vary for different examples and are given when explaining the examples below. The details of the four different dynamical systems are as follows:

- The 3D Lorenz model. The dynamical system is given by

$$\begin{aligned}\dot{x}_1 &= \sigma(x_2 - x_1)\\ \dot{x}_2 &= -x_2 - x_1x_3 + \alpha x_1\\ \dot{x}_3 &= x_1x_2 - \beta x_3\end{aligned} \tag{71}$$

  where $\sigma = 10, \alpha = 28, \beta = \frac{8}{3}$. This combination of parameters leads to a system that produces a chaotic solution with the famous butterfly pattern in state space. An adaptive Runge–Kutta solver is used to numerically integrate the dynamical system. The initial condition for the problem is taken to be [10, 10, 10]. The solution is recorded at time steps of $\mathrm{d}t = 0.05$ time units. The database consists of 100 snapshots taken *randomly* within the time range of 1000 time steps (or 50 time units). The δt for database A is taken to be 0.05 time units.
- The 9D Lorenz model. This model describes a system of nine quadratically coupled ODEs that exhibit chaotic behavior for a certain range of parameters. The 9D Lorenz model was first described and derived by Reiterer *et al.* [58]. The equations were originally derived as a model for the behavior of square convection cells and represent a truncated form of the Navier–Stokes equations. The 9D Lorenz model is given by

$$\begin{aligned}\dot{C}_1 &= -\sigma b_1 C_1 - C_2C_4 + b_3C_3C_5 + b_4C_4^2 - \sigma b_2 C_7\\ \dot{C}_2 &= C_1C_4 + -\sigma C_2 + c_4C_5 - C_5C_2 - \tfrac{1}{2}\sigma C_9\\ \dot{C}_3 &= -b_3C_1C_5 - b_4C_2C_2 - \sigma b_1 C_3 + C_2C_4 + b_2C_8^2\\ \dot{C}_4 &= -C_5C_2 + C_2C_3 - \sigma C_4 + C_4C_5 + \tfrac{1}{2}\sigma C_9\\ \dot{C}_5 &= \tfrac{1}{2}C_2^2 - 0.5C_4^2 - \sigma b_5 C_5\\ \dot{C}_6 &= C_9C_2 - C_9C_4 - b_6C_6\\ \dot{C}_7 &= -rC_1 - C_9C_4 + C_8C_5 - b_1C_7 + C_5C_8\\ \dot{C}_8 &= C_9C_2 + rC_3 - C_7C_5 - C_5C_7 - b_1C_8\\ \dot{C}_9 &= -rC_2 + rC_4 + 2(C_4 - C_2)C_6 + C_4C_7 - C_2C_8 - C_9\end{aligned} \tag{72}$$

  The parameters b_i are associated with the geometry of the convection cells and are defined as

$$b_1 := 4\left(\frac{1+a^2}{1+2a^2}\right), \quad b_2 := \left(\frac{1+2a^2}{2(1+a^2)}\right), \quad b_3 := 2\left(\frac{1-a^2}{1+a^2}\right)$$

$$b_4 := \left(\frac{a^2}{1+a^2}\right), \quad b_5 := 8\left(\frac{a^2}{1+2a^2}\right), \quad b_6 := \left(\frac{4}{1+2a^2}\right)$$

  The parameter a is a wave number parameter and is set to $a = 0.5$. There are two additional free parameters that dictate the behavior of the dynamical system, namely, the parameter r and σ. Reiterer *et al.* [58] studied the variation of these parameters in detail and concluded that for $\sigma = 0.5$ and $r \geqslant 14.22$ the flow exhibits a chaotic attractor. Therefore, we use these parameters when generating the ensemble for deriving our models and testing their performance.

MATLAB's adaptive RK-4 (Runge–Kutta 4) solver is used for numerical integration. The solution is recorded at intervals of $\mathrm{d}t = 0.05$ time units. The database consists of 100 snapshots taken from a time range of 1000 time steps (or 50 time units). The δt for database A is taken to be $\delta t = 0.05$ time units.

- A 5D nonlinear model (our 'in-house model'). This model is given by,

$$\begin{aligned}
\dot{Y}_1 &= q_1(Y_2 - Y_1) + Y_2 Y_3 Y_4\\
\dot{Y}_2 &= q_2(Y_1 + Y_2) - Y_1 Y_3 Y_4\\
\dot{Y}_3 &= -q_3 Y_3 + Y_1 Y_2 Y_4\\
\dot{Y}_4 &= -(q_4 + q_5 \sin(Y_5)) Y_4 + Y_1 Y_2 Y_3\\
\dot{Y}_5 &= -q_6 Y_5 + Y_1 Y_2 Y_3
\end{aligned} \tag{73}$$

The coefficients q_1–q_6 are given by $q_1 = 30$, $q_2 = 10$, $q_3 = 1$, $q_4 = 16$, $q_5 = 12$, $q_6 = 7.2$. The model not only contains cubic terms, but also trigonometric functions. The identification of such a dynamical system through nonlinear optimization would be an extremely tedious one. This model is used to demonstrate the flexibility of the RBF models to correctly identify the evolution of the original dynamical system. MATLAB's adaptive RK-4 (Runge-Kutta 4) solver is used for numerical integration. The solution is recorded at time interval of 0.01 time units. The database consists of 150 snapshots taken from a time range of 500 time steps. The δt for database A is taken to be 0.01 time units.

- The KS model. This model represents a solution that varies in one-dimensional space and time. The partial differential equation describing it is

$$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} - \frac{\partial^2 u}{\partial x^2} - \frac{\partial^4 u}{\partial x^4} \tag{74}$$

A solution to the above equation is sought over a domain of $x \in [0, 32\pi]$ with periodic boundary conditions. The above PDE with the periodic boundary conditions have no known analytic solution except for the non-trivial $u \equiv 0$ solution. Therefore, we seek a numerical solution for the above problem. The periodic boundary conditions enable the use of spectral methods to solve the problem using a Fourier decomposition. The advantage of using Fourier modes to perform the decomposition is that the modes immediately satisfy the boundary conditions. A total of 128 complex Fourier modes are used to solve the problem. An initial condition given by

$$u(x, t=0) = \cos\left(\frac{x}{8}\right)\left(1 + \sin\left(\frac{x}{8}\right)\right) \tag{75}$$

is chosen. Exponential time differencing with a fourth-order Runge-Kutta (ETDRK4) scheme is used to solve the system of ODEs. This problem with the initial and boundary conditions given above has been solved by Kassam and Trefethen [59]. We use the exact same procedure to generate the solution to the problem. The ETDRK4 scheme used for solving this problem has been derived and explained in detail in [59]. The KS equation is chosen as an example because it shares two important properties with the Navier–Stokes equation. The first property is that the low-dimensional form of the KS equation and the Navier–Stokes equations are the same, i.e. both equations lead to a quadratic system of ODEs. The second important property is the dimensionality of the system. More often than not, the solutions to fluid flow problems are high-dimensional, meaning they require a large number of degrees of freedom to correctly describe the solution. The KS equation in this regard also represents a high-dimensional system. Therefore, methods that work well with the low-dimensional form of the KS equation may be expected to also work for the Navier–Stokes equations. The solution is set up such that the integration time step is 0.01 time unit and the solution is recorded every five integration time steps (or every 0.05 time units).

For this example, a POD analysis is first performed on an ensemble of solutions. The POD analysis leads to a decomposition into spatial eigenfunctions $\boldsymbol{\psi}(x)$ and temporal coefficients $\boldsymbol{\zeta}(t)$. The evolution of the temporal coefficients $\boldsymbol{\zeta}(t)$ is now described by a quadratic system of ODEs. We are consequently interested in the evolution of the temporal coefficients, which are found from the RBF models. Since an additional step is performed prior to the construction of the RBF model (namely the POD analysis), we deal with this example in more detail at a later point in this section.

As part of our initial analysis, we use the first three examples to demonstrate that the RBF evolution models indeed represent the dynamics of the original system. To do this, we conduct four different tests:

1. Check if the RBF evolution model correctly reproduces the phase-space portraits of the original dynamical system.
2. Compare the Lyapunov exponents of the RBF evolution model with the original system.
3. Check the rate of divergence of the RBF evolution model with respect to the original system.
4. Compare the time derivatives of the original system with the time derivatives of the RBF models.

The first test is performed through visual inspection. Since the phase-space portraits of the systems considered here have strong features, reproduction of the phase-space portraits by the RBF model provides positive reinforcement to the method. The second and third tests can be quantified. Note that the second and third tests are quite different. The second test determines the rate at which trajectories that start out from *almost* the same initial condition diverge, whereas the third test establishes the rate at which the RBF model diverges from the original system for the exact same initial condition. The third test serves as a more robust test to determine if the RBF evolution models are indeed close to the original dynamical system. Since the systems that are being considered are chaotic, one could expect that the RBF models might diverge at a rate that is less than or equal to the largest Lyapunov exponent of the original system. The fourth example is used to check how well the RBF models are able to approximate the time derivatives of the original systems. One must note that for the fourth test, there is no interpolation involved. Rather, the solutions at different time steps are introduced into the RBF models and the resulting output (which is the time derivative for our case) is computed and compared with the time derivative as obtained from the original dynamical system.

We remind our readers that four different RBF models were considered. Two models each were constructed for database A and database B, respectively. To elucidate the different models, we label the two models formed from database A as ‘RBF-Model-A-Case1’ and ‘RBF-Model-A-Case2’. Similarly the models formed from database B will be called ‘RBF-Model-B-Case1’ and ‘RBF

Model-B-Case2'. To reiterate, the different models are formed as follows:

- RBF-Model-A-Case1: RBF model constructed using database A. The resulting model is a discrete map of the continuous dynamical system as described in Section 4.2.
- RBF-Model-A-Case2: Using RBF-Model-A-Case1, a continuous dynamical system is constructed based on the procedure outlined in Section 4.2.
- RBF-Model-B-Case1: RBF model constructed using database B using the procedure described in Section 4.1. Resulting model is a continuous dynamical model.
- RBF-Model-B-Case2: RBF model constructed using database B. This case focuses on constructing approximating dynamical systems that have a quadratic polynomial form. The resulting model can be written either in a polynomial form or in an RBF form.

All the experiments were performed using a multi-quadric radial basis function as given by

$$\Phi(r)=\sqrt{1+(\varepsilon\cdot r)^2} \tag{76}$$

Experiments with other basis functions were also conducted with results similar to the ones shown for the multi-quadric case. It should be noted that the multi-quadric basis function is a conditionally positive-definite function as opposed to a strictly positive-definite function. Even so, the interpolation properties still hold for this particular basis function. The scaling parameter ε is chosen based on the procedure defined in Section 5. For RBF-Model-A-Case1 and RBF-Model-B-Case1, the procedure outlined in Section 5.2 was used and the procedure outlined in Section 5.1 was used to compute the scaling parameter for RBF-Model-B-Case2. The scaling parameter of RBF-Model-A-Case1 was used for RBF-Model-A-Case2. Table I shows the scaling parameters used for the different RBF evolution models.

We would like to point out that the scaling parameter for the model 'RBF-Model-B-Case2' for the 5D system was not included in the table because it represents a specific case of modeling a quadratic system of ODEs. We provide evidence later that for this particular case, the RBF evolution model was always inherently unstable.

As mentioned earlier, our first test is to demonstrate that the RBF models are correctly able to construct the phase-space portraits, which we shall call 'PSP' for the sake of brevity. The PSPs are constructed by starting the different models at the same initial conditions. However, the time integration for each of the examples is different. Figures 1 and 2 show the PSP comparison of the 3D Lorenz model with the different RBF evolution models. One can easily see that the PSP correctly follows and forms the famous 'butterfly' pattern of the 3D Lorenz model. The PSPs are constructed based on the integration carried out over 40 time units (which corresponds to a total of 800 data points). It is important to note that the PSPs in Figures 1 and 2 have a jagged trajectory as opposed to a smooth one. This is because the solutions were extracted at discrete time steps.

Table I. The scaling parameters of the RBF models for the different cases and examples.

Model type	3D Lorenz	9D Lorenz	5D system
RBF-Model-A-Case1	0.04	0.2	0.68
RBF-Model-A-Case2	0.04	0.2	0.68
RBF-Model-B-Case1	0.02	0.2	0.15
RBF-Model-B-Case2	0.02	0.5	—

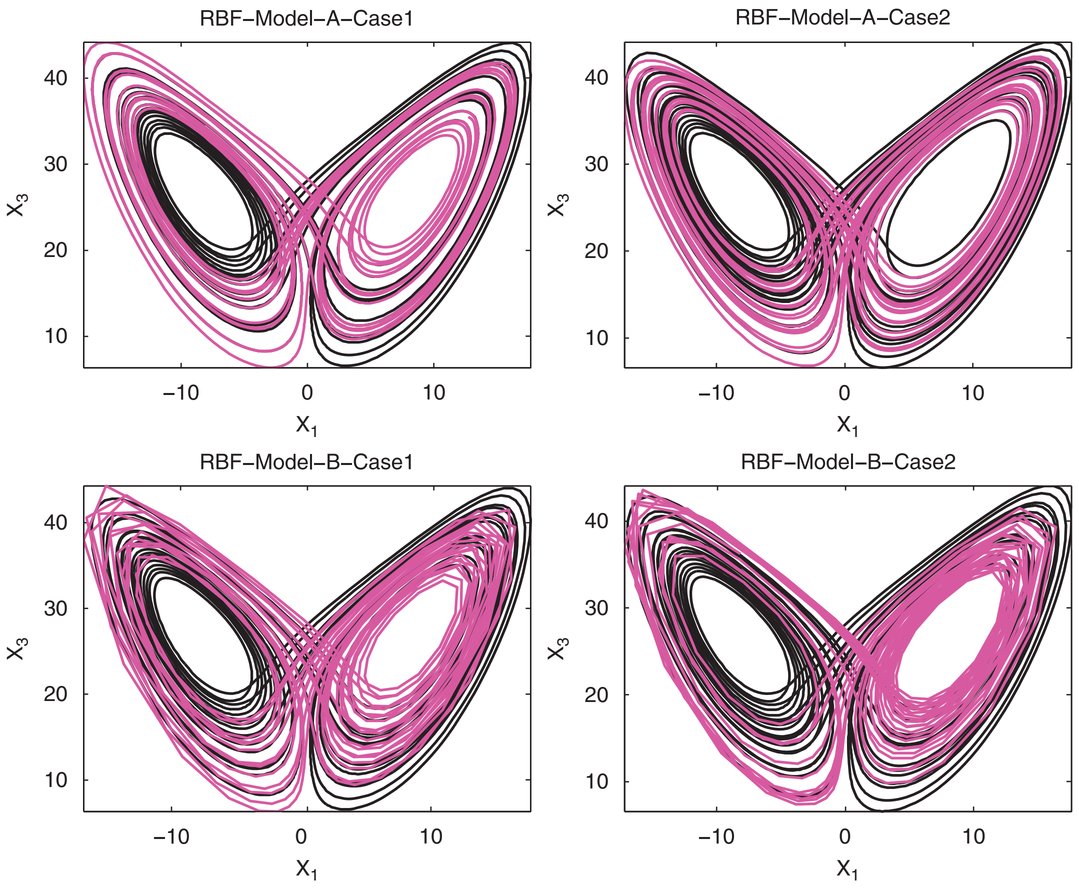

Figure 1. Phase-space portrait $x_1 - x_3$ of the 3D Lorenz system for the different RBF models. Black line denotes the reference solution. Time interval was 40 time units (800 data points).

For the 9D and the 5D Lorenz model, we perform comparisons of the PSP for a few selected variables. For the 9D Lorenz model, we select C_1, C_6, C_7 to construct sample phase-space portraits. Figures 3 and 4 show the PSP for the selected variables. In order to construct the PSP, the numerical integration was carried out for 150 time units (or 3000 data points). A longer integration time is required for this model because the RBF models follows the reference solution quite closely for nearly 100 time units.

Similar to the other two examples, Figures 5 and 6 show the PSP for the 5D system. For this example, the PSP are generated based on numerical integration performed over a period of 10 time units (or 1000 data points).

We now compare some fundamental properties of the dynamical systems. The Lyapunov exponents are perhaps the most fundamental quantities associated with a dynamical system that allow one to determine if the system is chaotic or not. Lyapunov exponents allow one to estimate the rate at which two trajectories will diverge when their initial conditions are separated by a very small distance. A dynamical system of dimension N has N Lyapunov exponents. Each one of

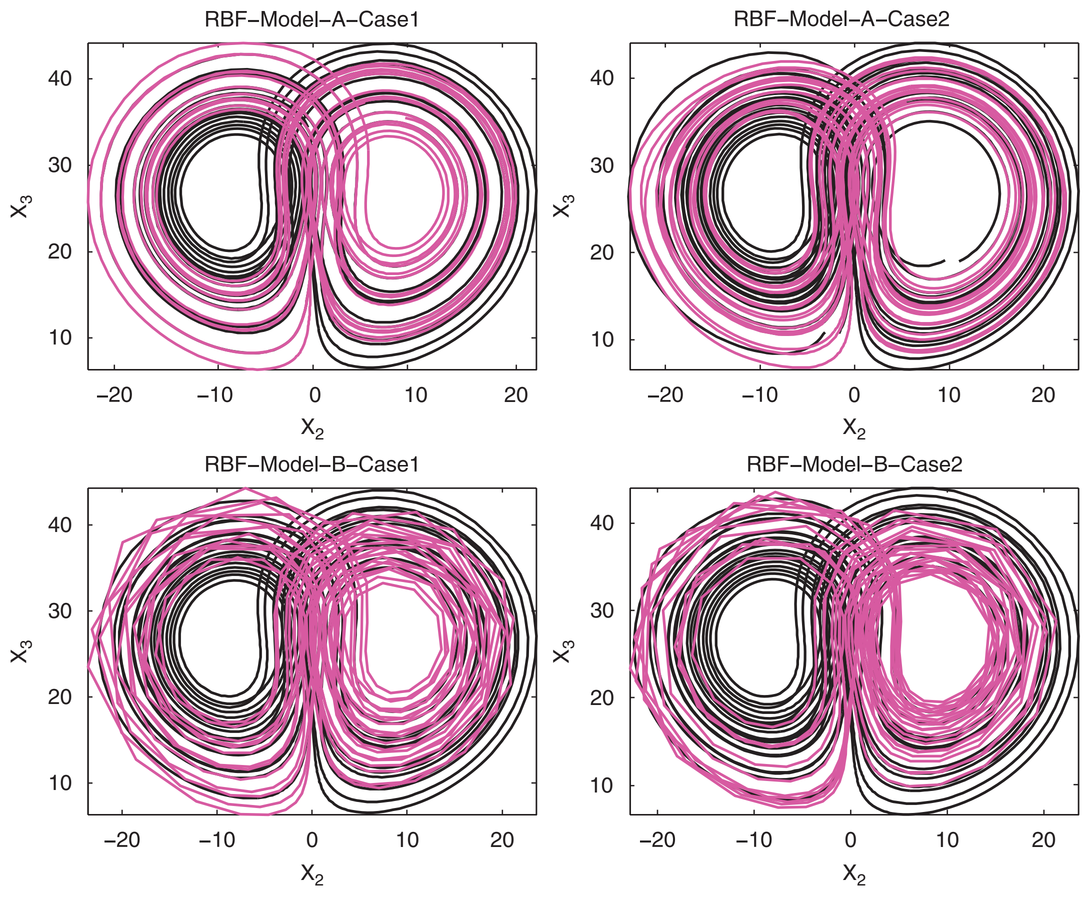

Figure 2. Phase-space portrait $x_2 - x_3$ of the 3D Lorenz system for the different RBF models. Black line denotes the reference solution. Time interval was 40 time units (800 data points).

these exponents corresponds to the rate at which the principle axis of a N dimensional ‘sphere’ in the nth direction increases or decreases with respect to time. For a system to be considered chaotic, one requires that at least one of the Lyapunov exponents be positive. Furthermore, when dealing with autonomous systems, one Lyapunov exponent must always be zero. For the first three examples, the Lyapunov exponents are given in Table II.

The method used for calculating the entire Lyapunov spectrum from the original dynamical system is explained in detail in the work of Wolf *et al.* [60], Shimada and Nagashima [61], Benettin *et al.* [62] and is often referred to as the ‘standard algorithm’ for computing the Lyapunov exponents. The standard algorithm requires that the system of ODEs be at least once differentiable. In our case, this condition is automatically met since the RBF functions are always differentiable. As a matter of fact, the multi-quadric function being used for our case is infinitely differentiable. Therefore, we are in a position to obtain the entire Lyapunov spectrum and compare it to the original dynamical system. Table II gives the exponents for the different examples being considered. Further, we also calculate the correlation dimension of the RBF evolution models and compare it to that of

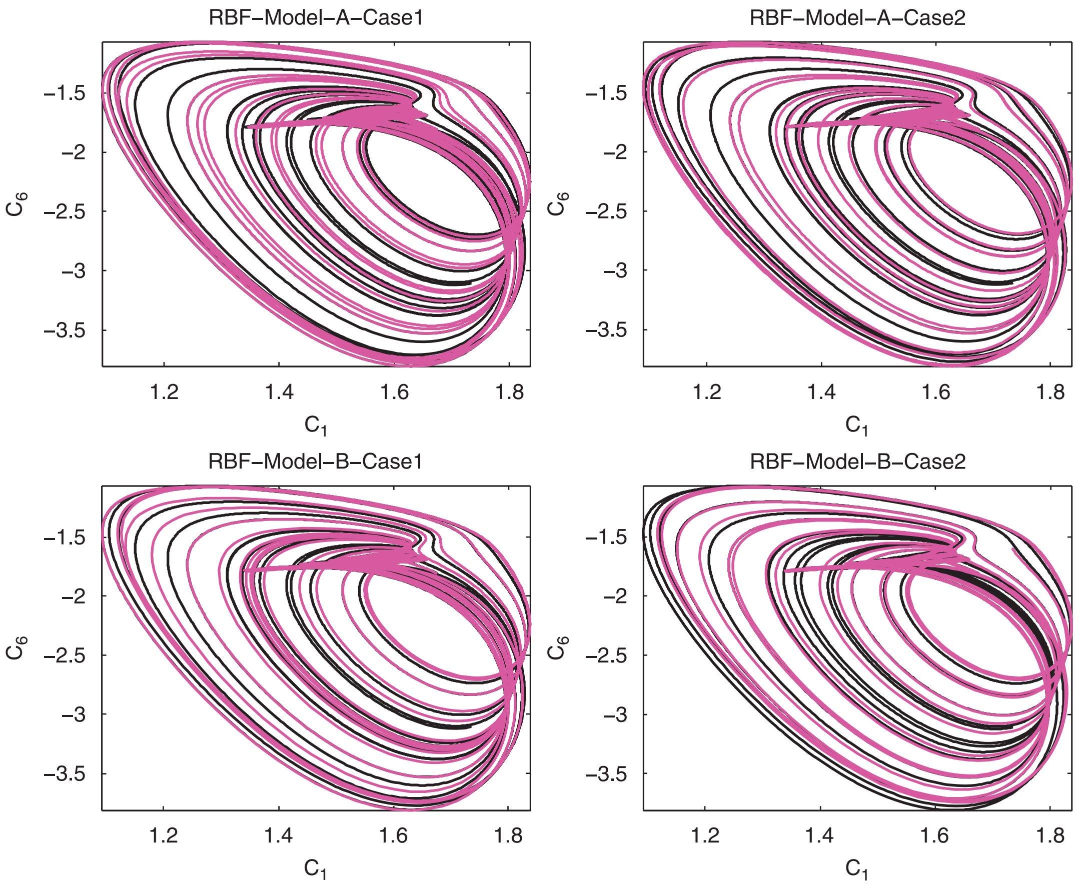

Figure 3. Phase-space portrait C_1-C_6 of the 9D Lorenz system for the different RBF models. Black line denotes the reference solution. Time interval was 150 time units (3000 data points).

the original system. The correlation dimension/Lyapunov dimension are closely related to the Lyapunov exponents of the dynamical system. The calculation of the correlation dimension from the Lyaunov exponents is addressed by Wolf [60] and is given by

$$\text{Corr. Dim} = j + \frac{\sum_{i=1}^{j} \lambda_i}{|\lambda_{j+1}|} \tag{77}$$

where λ_i are the Lyapunov exponents ordered from largest to smallest and j is defined as

$$\sum_{i=1}^{j} \lambda_i > 0 \quad \text{and} \quad \sum_{i=1}^{j+1} \lambda_i < 0 \tag{78}$$

We now compare the Lyapunov exponents of the different RBF models with the original system for the first three examples. Table III shows the exponents for the 3D Lorenz case. One can see

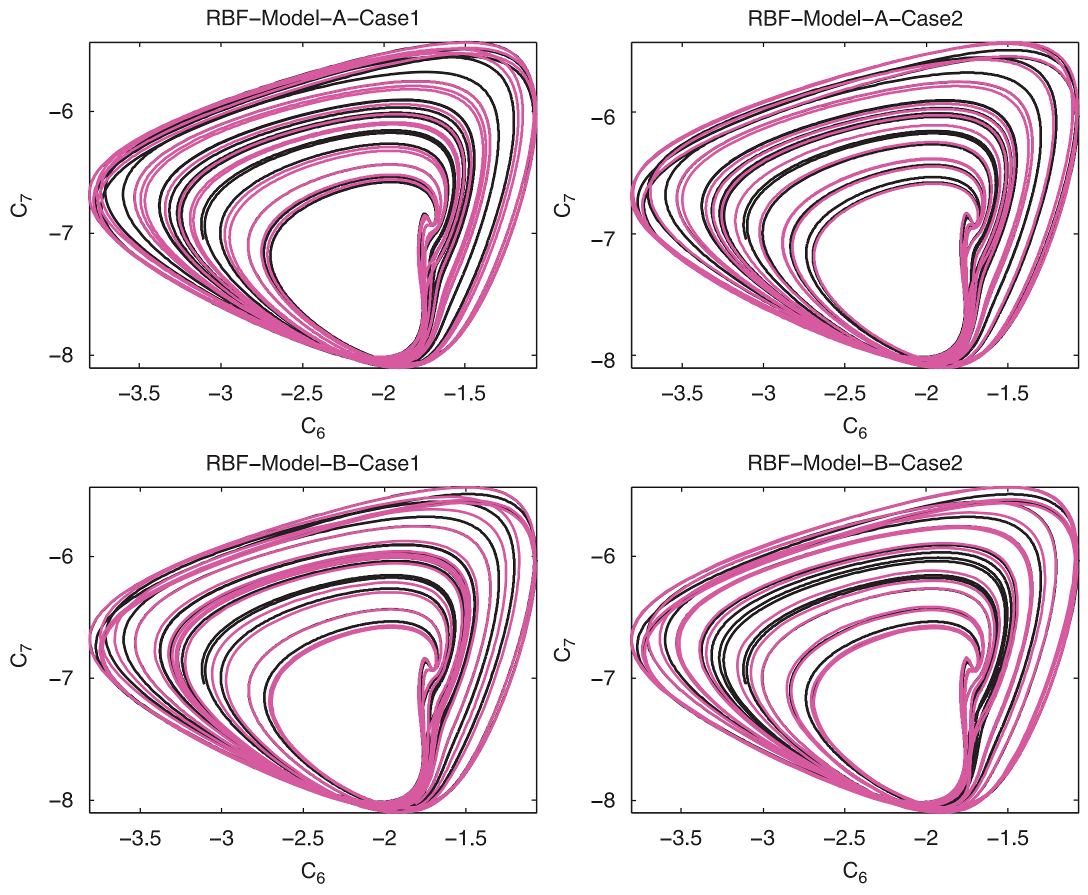

Figure 4. Phase-space portrait C_6-C_7 of the 9D Lorenz system for the different RBF models. Black line denotes the reference solution. Time interval was 150 time units (3000 data points).

that *all* the RBF evolution models show a spectrum that is quite close to the original dynamical system. It is also interesting to note that the fourth RBF model ‘RBF-Model-B-Case2’, which corresponds to the RBF model specific to quadratic systems seems to perform best. One can see a similar pattern in the results for the 9D Lorenz model and the 5D system. Table IV shows the comparison of the exponents for the 9D Lorenz case. A similar comparison of the third example, the 5D model is tabulated in Table V. Based on the cumulative comparison of the three examples, we find that among the four different RBF models, the RBF-Model-A-Case2 consistently seems to have a higher positive exponent than the other RBF models. This implies that RBF-Model-A-Case2 will tend to diverge faster than the other models. However, we point out that RBF-Model-A-Case2 is constructed as an extension to RBF-Model-A-Case1 wherein we are constructing a continuous model from a discrete RBF map using the linear multi-step methods. The linear multi-step method contains a local truncation error that is inherently embedded into the RBF system during the modeling phase. This error may explain the reason for the higher rate of divergence. We also

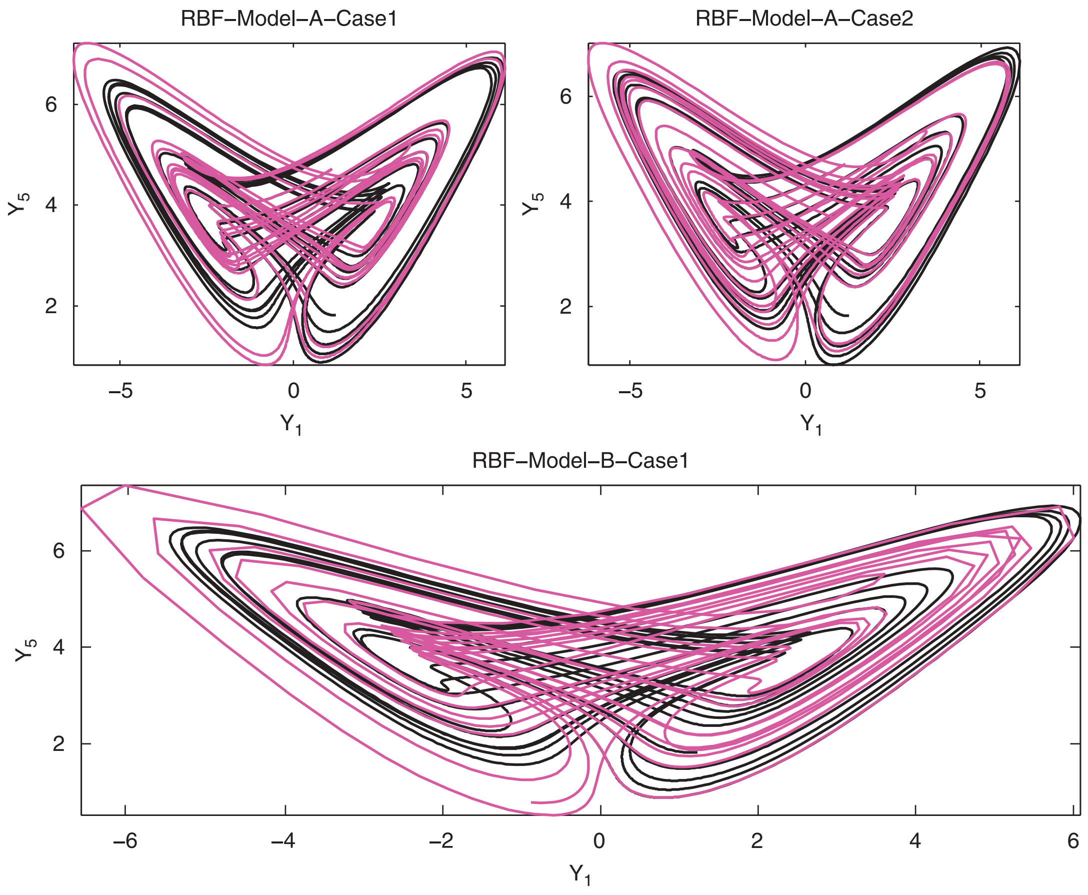

Figure 5. Phase-space portrait $Y_1 - Y_5$ of the 5D in-house system for the different RBF models. Black line denotes the reference solution. Time interval was 10 time units (1000 data points).

point out that the results associated with the RBF-Model-B-Case2 are not presented for the 5D in-house model as seen in Table V. This is because the 5D in-house model does not have a quadratic polynomial structure, while the RBF-Model-B-Case2 is constructed by assuming the original system being approximated has a quadratic polynomial form. We consistently found that the RBF models associated with RBF-Model-B-Case2 for the 5D model example were unstable, i.e. the solution became unbounded within a few integration time steps. As part of our analysis, it is also imperative to get an idea of how faithfully the RBF models follow the true dynamical systems and for how long. To do this quantitatively, we look at the 'time horizon' of the RBF models, i.e. the time beyond which the prediction of the RBF models breaks down (with respect to following the true dynamical system). We would like to point out that this is different from looking at the largest Lyapunov exponent of the RBF models. To get the prediction time horizon, the initial conditions for both the RBF models *and* the original dynamical system are the same. Table VI tabulates the time horizon for the different RBF models for the different examples.

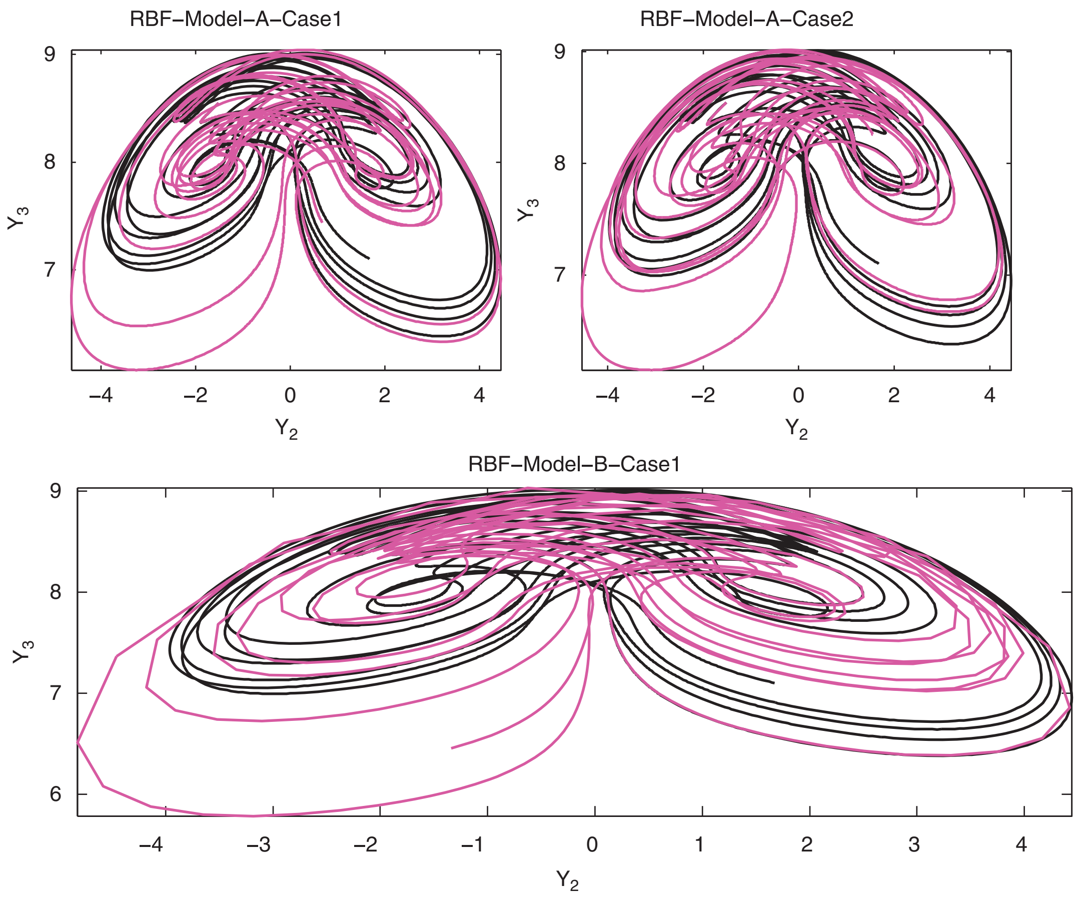

Figure 6. Phase-space portrait $Y_2 - Y_3$ of the 5D in-house system for the different RBF models. Black line denotes the reference solution. Time interval was 10 time units (1000 data points).

We define the time horizon as

$$\text{Time horizon} = \text{Max}_t \left\{ \frac{1}{N} \sum_{i=1}^{N} \frac{|x_i^{\text{rbf}}(t) - x_i^{\text{Orig}}(t)|}{|x_i^{\text{Orig}}(t)|} \right\} \overset{!}{=} 0.01 \tag{79}$$

where $x_i(t), i = 1, 2, \ldots, N$ represents the signal at time $t \cdot x_i^{\text{rbf}}$ represents the RBF solution and x_i^{Orig} represents the true solution as obtained by the original dynamical system. It is important to note that the RBF models are bound to diverge from the true solution. This is because even though the initial conditions were perfect, the RBF model in itself is an approximation to the original system. One can then view the RBF model in one of two ways. The first view could be that using the RBF model is equivalent to using the original system with a slightly different initial condition. This is bound to lead to a divergence whose rate is governed by the largest Lyapunov exponent. The other view is to say that the RBF model now represents a dynamical system whose parameters are slightly offset from the true/original dynamical system. For example, there exists

Table II. Lyapunov exponents for the three test cases.

Example	Lyapunov exponents
3D Lorenz system	[0.905, 0, −14.55]
9D Lorenz system	[0.0446, 0, −0.3943, −0.5949, −0.6193, −1.5109, −3.2776, −4.4920, −4.4964]
5D system	[3.2495, 0, −3.9166, −14.2000, −28.6552]

Table III. Comparison of the Lyapunov exponents and the correlation dimension of the different RBF models for the 3D Lorenz model.

Model	Lyapunov exponents	Corr. dim.
3D Lorenz	0.905, 0, −14.55	2.01
RBF-Model-A-Case1	0.8939, −0.0012, −14.7138	2.06
RBF-Model-A-Case2	0.9385, 0.0064, −14.3660	2.07
RBF-Model-B-Case1	0.9076, −0.0010, −14.5802	2.07
RBF-Model-B-Case2	0.9060, −0.0005, −14.5800	2.03

Table IV. Comparison of the Lyapunov exponents and the correlation dimension of the different RBF models for the 9D Lorenz model.

Model	Lyapunov exponents	Corr. dim.
9D Lorenz	0.0446, 0, −0.3943, −0.5949, −0.6193, −1.5109, −3.2776, −4.4920, −4.4964	2.10
RBF-Model-A-Case1	0.0461, 0.0003, −0.3700, −0.6020, −0.6374, −0.7473, −1.1194, −1.8186, −2.0960	2.14
RBF-Model-A-Case2	0.0504, 0.0029, −0.3580, −0.6369, −0.6748, −0.8716, −1.0056, −1.9578, −2.7408	2.15
RBF-Model-B-Case1	0.0451, 0.0003, −0.3845, −0.5763, −0.6417, −0.8329, −1.2450, −1.7521, −1.9524	2.2
RBF-Model-B-Case2	0.0450, −0.0001, −0.3576, −0.6217, −0.6417, −0.9321, −1.1935, −2.0310, −3.2541	2.07

three parameters σ, α, β that need to be defined for the 3D Lorenz model as given in (71). The RBF models could then be seen to represent a dynamical system for which perhaps one (or all) parameters σ, α, β are given by $\tilde{\sigma}=\sigma+\delta\sigma, \tilde{\alpha}=\alpha+\delta\alpha, \tilde{\beta}=\beta+\delta\beta$. In such a case, one again finds that starting from the same initial conditions, the original dynamical system and the modified dynamical system will diverge after some time. Nonetheless, these RBF models can be very useful when doing short-term prediction and their capabilities can easily be enhanced by suitably coupling them with measurement models through nonlinear Kalman filters.

Typical results associated with the RBF models for the time series also demonstrate that the RBF evolution models are indeed robust. Figure 7 shows a sample time series of the different RBF models to the reference time series (as indicated in black). It is again clear that RBF-Model-A-Case1

Table V. Comparison of the Lyapunov exponents and the correlation dimension of the different RBF models for the 5D Lorenz model. –**– indicates that the RBF model was unstable.

Model	Lyapunov exponents	Corr. dim.
5D system	3.2495, 0, −3.9166, −14.2000, −28.6552	2.80
RBF-Model-A-Case1	3.2109, 0.0016, −4.2289, −15.4800, −27.3040	2.76
RBF-Model-A-Case2	3.3593, 0.0054, −3.8367, −13.4831, −27.3766	2.89
RBF-Model-B-Case1	3.2442, 0.0014, −3.9819, −14.5370, −28.9859	2.75
RBF-Model-B-Case2	– ** –	– ** –

Table VI. The time horizon (in time units) of different RBF models for the different examples.

Model type	3D Lorenz	9D Lorenz	5D system
RBF-Model-A-Case1	≈ 5	≈ 100	≈ 2
RBF-Model-A-Case2	≈ 3	≈ 75	≈ 1
RBF-Model-B-Case1	≈ 5	≈ 100	≈ 2
RBF-Model-B-Case2	≈ 7	≈ 100	–** –

The time horizon represents the time when the relative error between the RBF solution and the original solution is 1%.

and RBF-Model-B-Case1 do indeed perform quite well. Similar results are seen in Figures 8 and 9. Figure 8 shows the time series for the 9D Lorenz model while Figure 9 presents the time series plots for the 5D system.

We now look at our final test to establish that the RBF models are indeed good representations of the original dynamical system. We study the capability of the RBF models to correctly predict the time derivatives of the multivariate signals. The RBF models are provided with the value of the signals that lie on the attractor and compute the time derivative of the signals at that instance in time. These results are compared with the time derivatives computed from the original systems. The average error between the RBF models and the true solution is compared on an ensemble of data. The error is computed as

$$\text{Error} = \left\langle \frac{1}{N} \sum_{i=1}^{N} \frac{|\dot{x}_i^{\text{rbf}}(t) - \dot{x}_i^{\text{Orig}}(t)|}{|\dot{x}_i^{\text{Orig}}(t)|} \right\rangle \tag{80}$$

where $\dot{x}_i^{\text{rbf}}$ represents the RBF solution and $\dot{x}_i^{\text{Orig}}$ represents the time derivatives as obtained by the RBF model and the original dynamical system. $\langle\rangle$ represents an ensemble average taken over arbitrary instances in time. These results are tabulated for the different RBF models and different examples in Table VII. It is interesting to note that all the models produce an accurate approximation to the true time derivatives. Studying the errors in Table VII, it is interesting to see that the RBF models RBF-Model-B-Case1 and RBF-Model-B-Case2 are superior to the former two models. This result is easily explained based on the fact that these models are created with the assumption that the time derivatives are already known at certain instances in time along with the signal itself. We also recall that the comparison of the time derivative for the RBF model RBF-Model-A-Case1 does not apply since this model is a discrete evolution map. However, its extension,

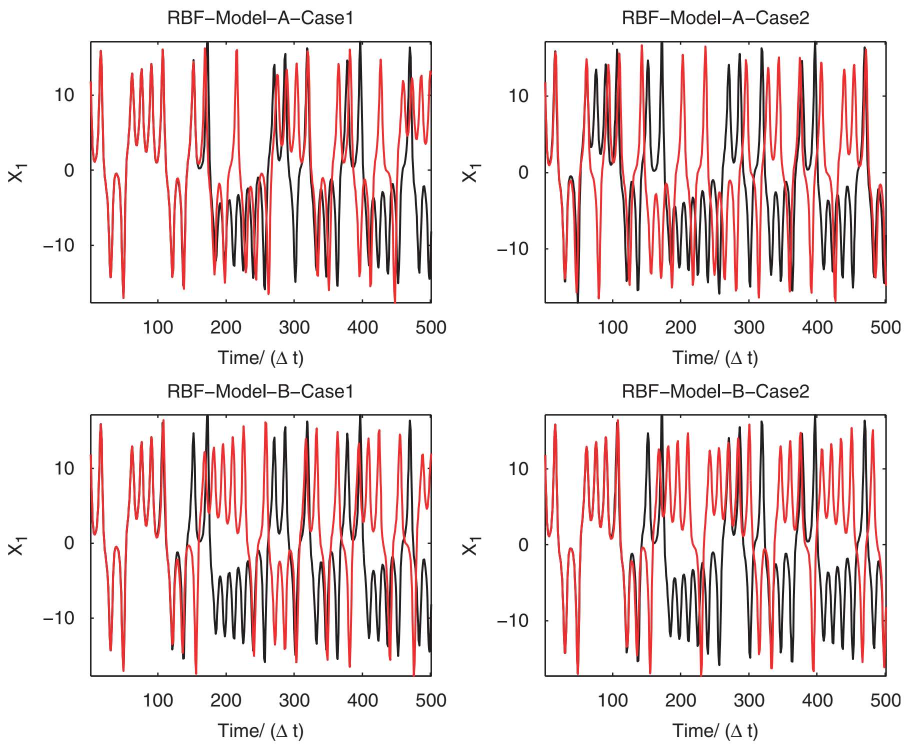

Figure 7. Time series of X_1 of the 3D Lorenz system for the different RBF models. Black line denotes the reference solution.

RBF-Model-A-Case2 is seen to produce very good approximations to the time derivatives with an average error of around 0.1% for all the examples. For the two quadratic system considered (the 3D Lorenz and the 9D Lorenz), we find that models associated with RBF-Model-B-Case2 give the best results. It is also seen that for the 3D Lorenz case, an error very close to machine precision is obtained.

To summarize, we performed four different tests on three different examples to get a quantitative answer to whether the RBF evolution models were suitable candidates to modeling chaotic nonlinear dynamical systems given only sparse time series information. Based on the four tests, one can conclude that the models are not only valid, but robust and do represent the dynamics of the original system.

As a final example, we now apply the RBF evolution model to approximate the KS equation. The details of the KS equation and the procedure for computing the numerical solution were discussed in the earlier part of this section. Figure 10 shows a sample solution obtained by the Fourier-spectral method. The spatio-temporal chaos in the solution is seen in the form of different ‘waves’ splitting and merging at different instances in time and space. Before proceeding to the construction of the RBF evolution model, we perform a POD decomposition on the solution of KS equation. The POD analysis decomposes the solution into a set of spatial functions $\psi(x)$ and corresponding to each of these functions there exists a coefficient $\zeta(t)$, which depends only on

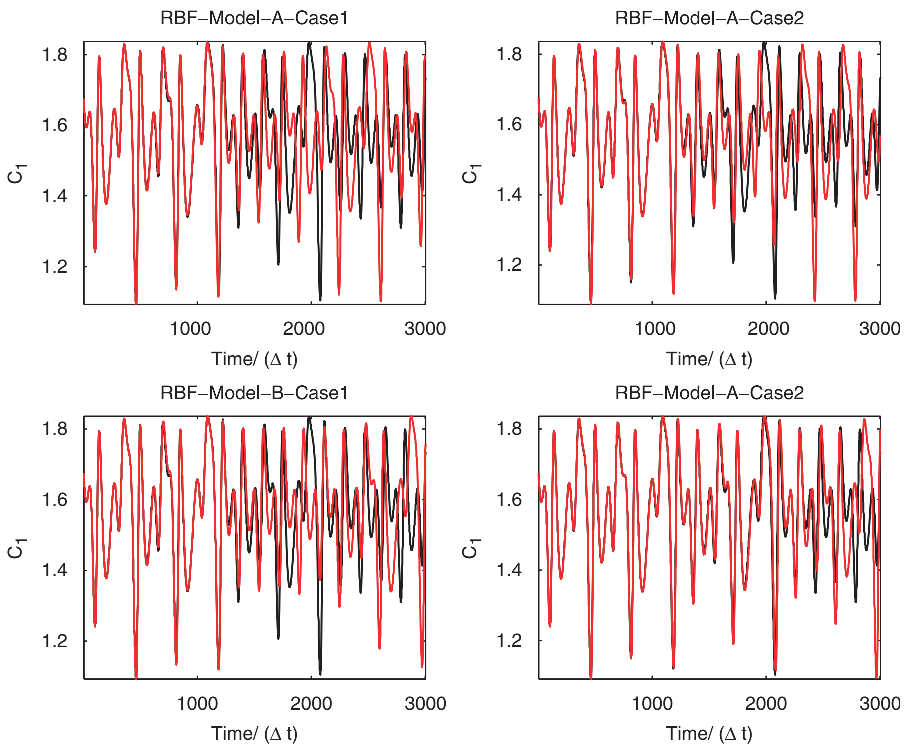

Figure 8. Time series of C_1 of the 9D Lorenz system for the different RBF models. Black line denotes the reference solution.

time. The details of the POD analysis are discussed in Section 2 of this paper. Once the POD analysis is done, the original KS equation, which is given by

$$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} - \frac{\partial^2 u}{\partial x^2} - \frac{\partial^4 u}{\partial x^4}$$

is converted to a system of ODEs by projecting the above equation onto the POD basis functions $\psi(x)$. The system of ODEs is given by

$$\frac{\mathrm{d}\zeta_k}{\mathrm{d}t} = \sum_{i=1}^{N}\sum_{j=1}^{N} \mathscr{H}_{kij}\zeta_i\zeta_j + \sum_{i=1}^{N} \mathscr{G}_{kj}\,\zeta_j, \quad k=1,2,\ldots,N \tag{81}$$

where the elements of $\mathscr{H}$ and $\mathscr{G}$ are given by

$$\begin{aligned}\mathscr{H}_{kij} &= -\int \psi_k \cdot \left(\psi_i \cdot \frac{\mathrm{d}\psi_j}{\mathrm{d}x}\right)\mathrm{d}x \\ \mathscr{G}_{kj} &= -\int \psi_k \cdot \left(\frac{\mathrm{d}^2\psi_j}{\mathrm{d}x^2} + \frac{\mathrm{d}^4\psi_j}{\mathrm{d}x^4}\right)\mathrm{d}x\end{aligned} \tag{82}$$

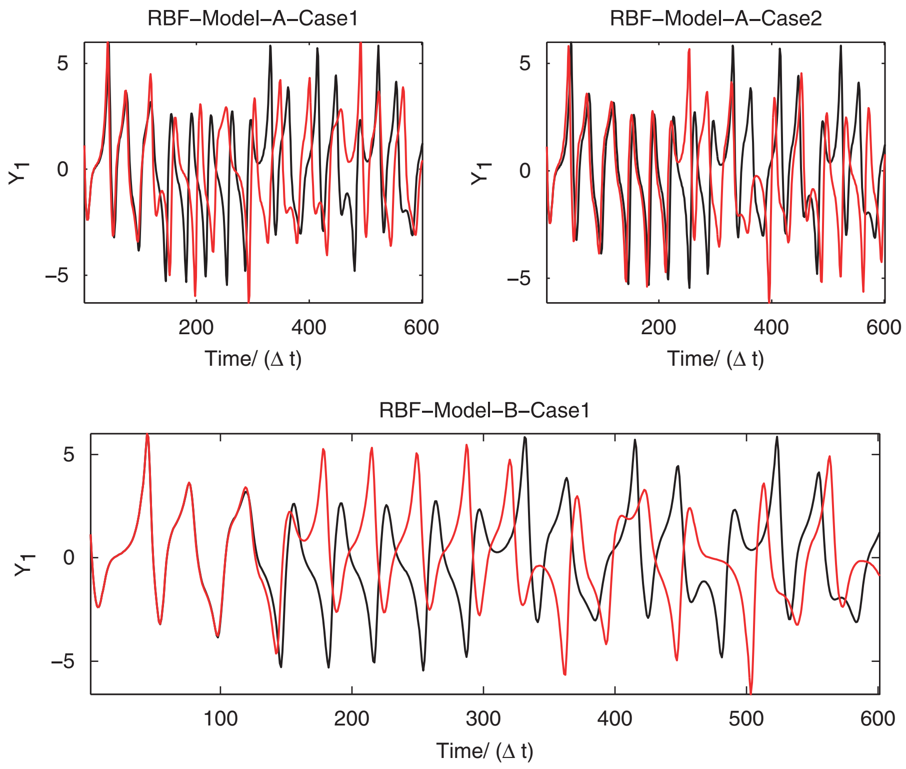

Figure 9. Time series of Y_1 of the 5D system for the different RBF models. Black line denotes the reference solution.

Table VII. Error associated with time derivatives as computed by using RBF evolution models.

Example	RBF-Model-A-Case1	RBF-Model-A-Case2	RBF-Model-B-Case1	RBF-Model-B-Case2
3D Lorenz	–NA–	4.2656×10^{-3}	1.3393×10^{-10}	$\approx 10^{-16}$
9D Lorenz	–NA–	1.8136×10^{-5}	1.6612×10^{-6}	8.4253×10^{-9}
5D Lorenz	–NA–	4.4281×10^{-3}	3.7020×10^{-5}	– ** –

The error is computed using (80). –**– indicates that we could not find a stable RBF model for the specified case. –NA– indicates not applicable.

It can now be seen that the solution of the KS equation can be found by solving a quadratic system of ODEs. The POD analysis requires an initial ensemble on which the decomposition is performed to extract the basis functions. The initial ensemble is obtained from the spectral solution. It is important to note that since this equation does not have a closed-form solution, we use the spectral solution as our reference solution. The initial ensemble consists of 1000 snapshots taken at an interval of 1 time unit. Figure 11 shows the eigenvalue plot for the POD modes. By requiring that a minimum of 99% of energy is captured, we find that the 75 POD modes

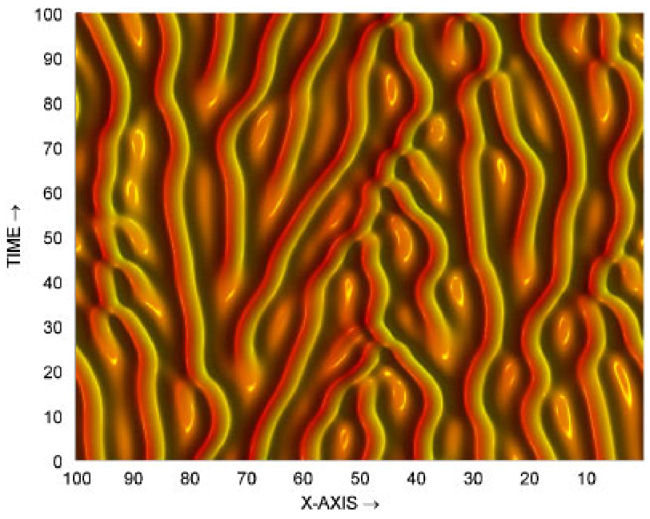

Figure 10. Solution of the KS equation as computed using Fourier-spectral method.

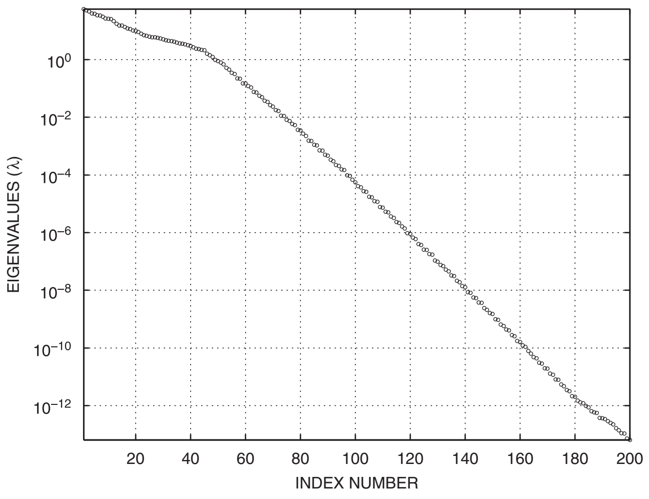

Figure 11. POD eigenvalues of the KS equation.

satisfy this condition. Therefore, the approximation to the solution is truncated to 75 modes. The approximation is given by

$$u(x,t)=\sum_{i=1}^{i=75}\zeta_i(t)\psi_i(x) \tag{83}$$

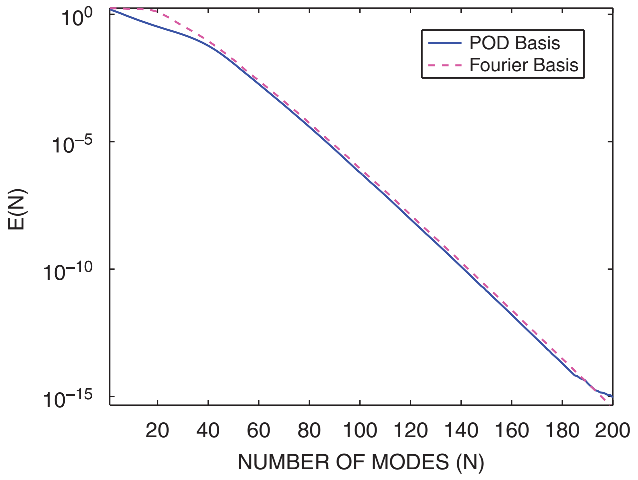

Figure 12. Mean-squared reconstruction error $E(N)$ as computed using (85) for POD modes and Fourier modes.

It is important to note that our analysis reveals that the POD analysis does not recover the Fourier modes as the spatial eigenfunctions. To explain this, we first note that the POD eigenfunctions are Fourier modes only if the correlation matrix is

$$\mathbf{R}(\mathbf{x},\mathbf{x}')=\mathbf{R}(\mathbf{x}-\mathbf{x}')=\mathbf{R}(r)=\sum_{k=-K}^{K} c_k \mathrm{e}^{i\alpha_k(\mathbf{x}-\mathbf{x}')} \tag{84}$$

From the theory of POD, one can show that if the equations are translationally invariant and the solution satisfies periodic boundary conditions, then the optimal modes must be Fourier modes. However, for a limited ensemble spanning a finite time period, one can intuitively expect that a different set of basis functions might provide a better representation of the ensemble. To demonstrate this, one can look at the mean-squared error associated with the reconstruction of the initial ensemble, as given by

$$E(N)=\frac{1}{P}\sum_{p=1}^{P}\left(u(x,t_p)-\sum_{n=1}^{N}a_n(t_p),\Gamma_n(x)\right)^2 \tag{85}$$

where $a_n/\Gamma_n(x)$ could be the Fourier coefficients/Fourier modes or POD coefficients/POD modes. Figure 12 shows the error associated with the number of terms used for reconstruction when Fourier modes and POD modes are used. It can clearly be seen that the POD modes have a smaller reconstruction error as compared with the Fourier expansion. However, the difference in error is seen to be small. One can expect that if the ensemble was extremely large, then the POD modes would indeed converge to Fourier modes. For the purpose of this paper, the POD eigenfunctions are always used to perform any and all analysis for the KS solution.

The evolution Equation (81) for the coefficients now represents a low-order model of the original system. One must note that the higher-order modes, i.e. modes with indices greater than 75 are now neglected based on the assumption that they contribute little to the overall dynamics of the system. However, neglecting the higher-order modes could render the system of ODEs unstable. Eddy-viscosity type models have been suggested to model the unresolved modes and stabilize the system. These procedures have been elaborated in the work of Noack *et al.* [15], Rempfer and Fasel [12], Sirisup and Karniadakis [17] and others. For our case, we apply a simple modification to the eddy-viscosity model, as proposed by Rempfer and Fasel [12], to account for the unresolved modes. An additional linear term is added as a damping term. The new equation now looks as

$$\frac{\mathrm{d}\zeta_k}{\mathrm{d}t}=\sum_{i=1}^{N}\sum_{j=1}^{N}\mathscr{H}_{kij}\zeta_i\zeta_j+\sum_{i=1}^{N}(\mathscr{G}_{kj}+\mathscr{D}_{kj})\zeta_j \tag{86}$$

where the term $\mathscr{D}$ is the additional damping term. To obtain the elements of the matrix $\mathscr{D}$, we apply the above equation to the coefficients obtained within the ensemble, i.e. we set $\zeta_k(t)=\zeta_k^{\mathrm{EN}}(t)$, $k=1,2,\ldots,N$ and multiply both sides of the equation with $\zeta_m^{\mathrm{EN}}(t)$ and integrate over the ensemble time. This gives an explicit relation to the matrix $\mathscr{D}$ as

$$\mathscr{D}_{km}=\frac{1}{\vartheta_m}\left(\left\langle\frac{\mathrm{d}\zeta_k^{\mathrm{EN}}}{\mathrm{d}t},\zeta_m^{\mathrm{EN}}\right\rangle-\sum_{i=1}^{N}\sum_{j=1}^{N}\mathscr{H}_{kij}\langle\zeta_i^{\mathrm{EN}}\zeta_j^{\mathrm{EN}},\zeta_m^{\mathrm{EN}}\rangle-\mathscr{G}_{km}\right) \tag{87}$$

where ϑ_j are the eigenvalues associated with the POD eigenfunctions. We would like to point out that this procedure is equivalent to computing the linear terms in (81) using a least-squares method. The POD-Galerkin model for the KS equation is now given by (86).

We now turn our attention to the RBF evolution model. As before, we can consider two different scenarios. The two scenarios represent the way the input–output data is available. These scenarios were elaborated in detail in the beginning of this section. Database A corresponds to a case where the input/output is given by $(\boldsymbol{\zeta}(t)/\boldsymbol{\zeta}(t+\delta t))$ and database B presents input/output as $(\boldsymbol{\zeta}(t)/\dot{\boldsymbol{\zeta}}(t))$. We construct an RBF evolution model for each of these databases and compare our results with the Fourier-spectral solution (also considered as reference solution) and POD-Galerkin solution. For database A, we construct a discrete evolution map (which we shall call 'RBF-Model-Discrete') and for database B we construct a general RBF-evolution map (which we shall call 'RBF-Model-Continuous'). The δt for database A is taken to be 0.01 time units. The input parameters in the database are taken to be the POD coefficients obtained from the initial ensemble. This corresponds to the database consisting of 2000 'snapshots' (please refer to the beginning of the section on how 'snapshots', 'database' are defined and the explanation of other notations).

All solutions for the RBF evolution and the POD-Galerkin models are performed outside the initial ensemble used for construction of the model or the construction of the POD modes. As a first test for this example, we first check to see if the different POD coefficients $\boldsymbol{\zeta}(t)$ as obtained by the different RBF models have a standard deviation that is consistent with the reference solutions. Since each POD coefficient carry the information about the amount of energy present in each of the POD basis functions/modes $\psi(x)$, the standard deviations provide an envelope in which the solutions should lie for any coefficients and hence have a certain amount of energy. Figure 13 shows the standard deviations of the 75 POD modes as obtained from the two RBF models considered. It is clearly visible that both the models have standard deviations quite close to the reference solution. This result indicates that, on average, the RBF models are able to distinguish the amount of energy present in the different scales of motion. We finally compare the solutions as obtained

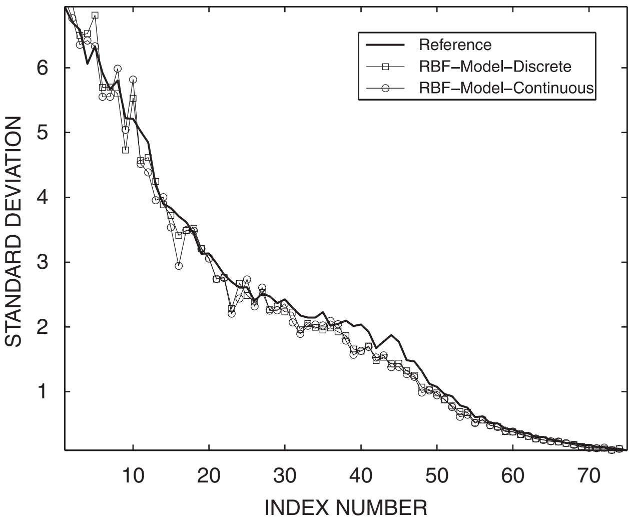

Figure 13. Standard deviations of the two RBF models as compared with the reference solution.

by the RBF models to the reference solution. This is done in two parts. We first compare the time series of the POD coefficients over a period of 100 time units. Figure 14 shows the comparison of the reference solution to the discrete RBF model and the time series for the continuous RBF model is seen in Figure 15. It is seen from Figures 14 and 15 that the solution diverges quickly to a different trajectory. The time horizon for the divergence is computed to be approximately 2 time units for both the models, indicating that the models represent good approximations only for a short time horizon. The divergence of the solution is not completely unexpected. The solution of the original system (KS equation) represents deterministic chaos. Combining with the fact that the POD-Galerkin system of ODEs representing the original system is also high dimensional (75-dimensional system), one is dealing with a system that is quite sensitive to small perturbations in the system. The POD-Galerkin system already represents an approximation to the original KS equation, i.e. there exists a truncation error with respect to the number of modes used and modeling error associated with accounting for the unresolved POD modes. Therefore, the POD-Galerkin system will also diverge to a different solution after finite time steps. Since the RBF systems represent models that approximate the original POD-Galerkin system, one should expect that the RBF systems will diverge much faster than the POD-Galerkin system.

However, despite the rapid divergence of the solution, when one compares the spatio-temporal solution of the different RBF models with the reference solution, one sees that the overall solution obtained by the models still reflects the structures seen in the true/reference solutions. Figure 16 shows a comparison of the different RBF models. One can see that the long meandering structure that the true solution exhibits is also seen in the two RBF models. The merging and splitting features of the meandering structures are also clearly visible in the two RBF models. Further, if one looks at the beginning of the time series, one sees that the structures start out similar to one another and then diverge out to different solutions. This result, therefore, indicates that the RBF models are able to recognize the underlying dynamics of the KS equation.

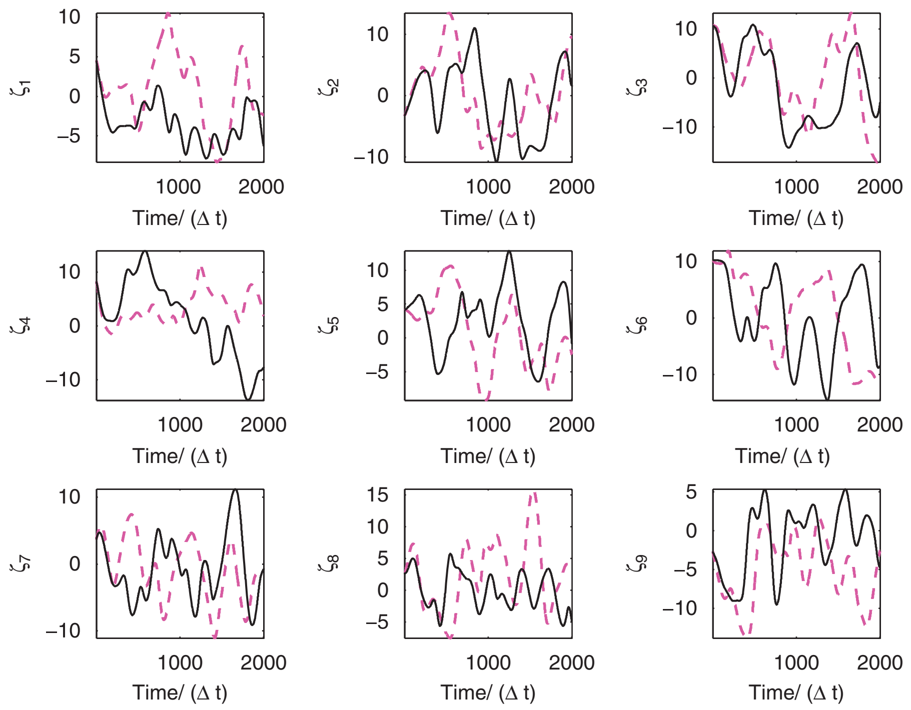

Figure 14. Comparison of the time series of the first nine POD coefficients for the discrete RBF evolution map with the reference solution. Solid line is the reference solution.

7. CONCLUSIONS

This paper has explored the idea of using global radial basis functions (RBFs) in the construction of nonlinear dynamical systems. Although RBFs have been used in the form of neural networks, the method has essentially remained a black-box technique, which restricted the usage of RBFs to just time series prediction and rendered it ineffective at analyzing the dynamical system itself. This paper approaches the use of RBFs from the perspective of constructing dynamical systems, discrete and continuous, based on time series data available from the original dynamical system. Based on the type of data available, two different databases were constructed. Database A designates cases when pairs of discrete time series data were given. Database B denotes cases when the time series and its time derivatives were given.

Based on database A, it was shown that one could construct a discrete RBF evolution map. The map uses the coefficients at a particular instance in time t and outputs the value of the series at time $t+\delta t$, where δt is the time separation between the pairs of data in database A. Based on the discrete RBF evolution map and linear multi-step methods, a method was developed that allows us to produce a continuous dynamical systems model from discrete data, i.e. database A. The discrete evolution map was labeled 'RBF-Model-A-Case1' and 'RBF-Model-A-Case2' for the two cases, respectively. Similarly for database B, two different methods were described that allow us to construct continuous RBF systems, i.e. the model is given the value of the multivariate coefficients

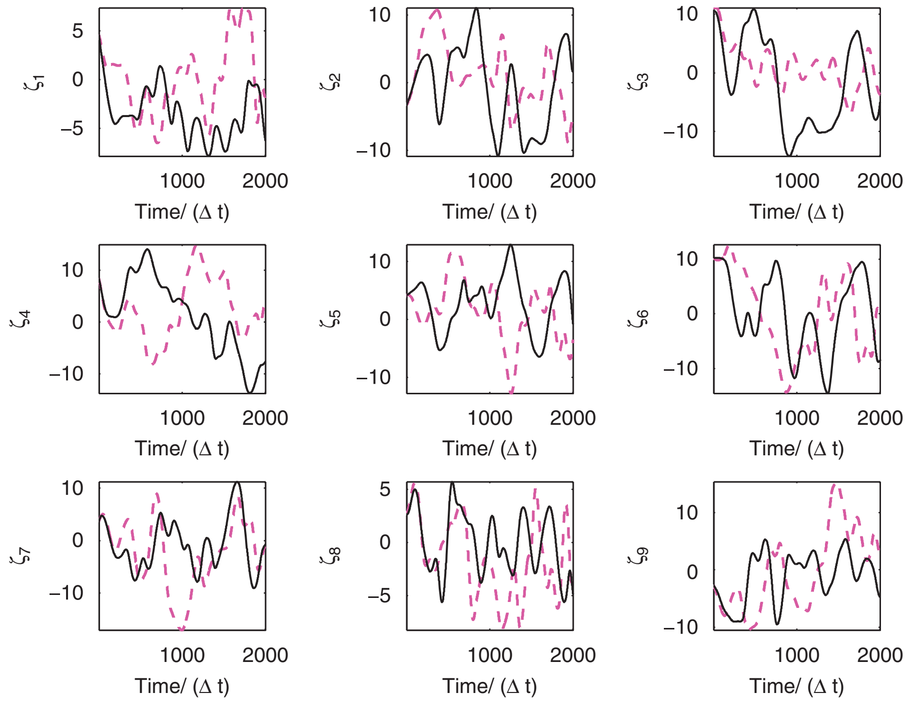

Figure 15. Comparison of the time series of the first nine POD coefficients for the continuous RBF model with the reference solution. Solid line is the reference solution.

at time t and it outputs the time derivatives of the coefficients at time t. The RBF system can be treated as any system of ODEs and numerically integrated to get the value of the coefficients at different instances in time. The first method for database B constructs a RBF evolution model that can be used, in principle, for approximating any nonlinear dynamical system. This evolution model was labeled 'RBF-Model-B-Case1'. The second method allows one to approximate dynamical systems that have a quadratic polynomial form. This type of model is of particular importance for us, since our motivation is to construct dynamical systems that arise from the Navier–Stokes equations. The RBF model of this type was labeled 'RBF-Model-B-Case2'.

In order to demonstrate the different RBF models, three examples were considered. The three examples consist of systems of ODEs that exhibited chaotic behavior. The 3D Lorenz model, the 9D Lorenz model and a 5D in-house model. Four different comparison criteria were considered for the RBF models. The first test was to check if the RBF models were able to construct the phase-space portraits of the original dynamical systems. It was shown that all models performed extremely well for phase-space reconstruction. The second test was to compare the Lyapunov spectrum of the RBF systems with that of the original system. It was seen for this test that all the models had Lyapunov spectra that were close to the original dynamical system. The largest Lyapunov exponent and the correlation dimensions of the different models were also found to be consistent with the reference system. The third test compared the error associated with the RBF models when computing the time derivatives of the time series coefficients. For this case, RBF-Model-A-Case1 was not part of the test analysis since it is a discrete evolution map. However,

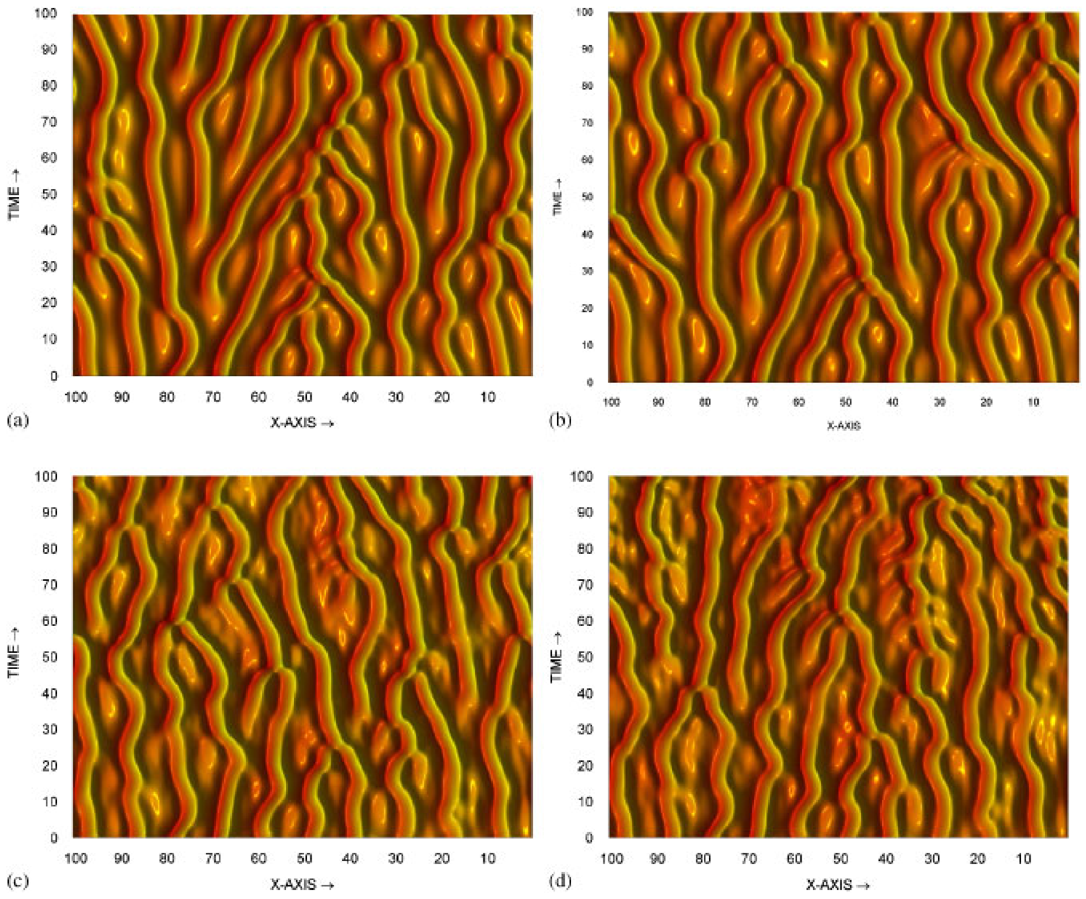

Figure 16. (a) Actual KS solution; (b) solution from POD-Galerkin model; (c) solution from the discrete RBF evolution map; and (d) solution from the continuous RBF evolution model.

it was shown that all the RBF models had an average approximation error of the order of 0.1%. For the fourth test, the time horizon over which the RBF models faithfully followed the true system was computed. It was seen that the models diverged at a rate comparable with the largest Lyapunov exponent. Overall, the combination of the four different tests confirmed that the RBF models were indeed dynamically consistent models.

To demonstrate the use of the RBF models for capturing spatio-temporal chaos, the KS equation was considered as a candidate example. This example was chosen because the low-dimensional model formed by the partial differential equation has the same quadratic polynomial form as the low-dimensional form of the Navier–Stokes equations. A Fourier-spectral decomposition with 256 modes (128 complex modes) was used to solve the problem and this solution was treated as the reference solution. For this case, before the RBF models were constructed, a POD analysis was first performed on the KS solution. This yielded a low-dimensional model that required 75 terms for approximating the solution. The POD analysis decomposes the flow into a set of spatial functions and corresponding to each function is a temporal coefficient, which is a function of time only.

RBF models were then constructed to model the evolution of the POD temporal coefficients. For this case, two different RBF models were constructed, a discrete evolution map and a continuous dynamical system. The standard deviations of the coefficients were compared with the reference solution. The results showed very good agreement with the reference. Time series comparison of the two models with respect to the reference solution were also shown for 2000 time steps. It was observed that the RBF models diverged to a different solution at a very rapid rate. The computed time horizon for the RBF models was found to be approximately 2 time units. However, when the entire spatio-temporal solution was constructed, it was observed that the underlying dynamics of the system were very well represented by both RBF models.

Details on how to compute the optimal scaling parameters for the different RBF models were also discussed and the different scaling parameters used for the different examples and models were tabulated. Based on the four different examples, and the different tests performed, the use of RBF to model continuous dynamical system can be considered a valid procedure. The advantages of using RBF models to approximate *any* nonlinear multivariate dynamical system are clear. Further, based on the methods discussed in this paper, one can construct continuous dynamical systems irrespective of the form in which the data are available. Special attention was paid to the idea of constructing RBF models that represent quadratic systems of ODEs. One of the primary motivations for developing these types of models was to introduce an alternative to the traditional Galerkin approach of constructing dynamical systems. When dealing with several hundred modes, the lower computational cost associated with the RBF models when compared with the Galerkin approach represents an attractive feature.

REFERENCES

1. Berkooz G, Holmes P, Lumley JL. The proper orthogonal decomposition in the analysis of turbulent flow. *Annual Review of Fluid Mechanics* 1993; **25**:539–575.
2. Holmes P, Lumley JL, Berkooz G. *Turbulence, Coherent Structures, Dynamical Systems, and Symmetry*. Cambridge University Press: Cambridge, New York, 1996.
3. Lumley JL. *Stochastic Tools in Turbulence*. Academic Press: London, U.K., 1970.
4. Aubry N, Holmes P, Lumley J, Stone E. The dynamics of coherent structures in wall region of a turbulent boundary layer. *Journal of Fluid Mechanics* 1988; **192**:115–173.
5. Sirovich L. Turbulence and the dynamics of coherent structures. Part I: coherent structures. *Quarterly of Applied Mathematics* 1987; **45**(3):561–571.
6. Sirovich L, Everson R. Management and analysis of large scientific dataset. *International Journal of Supercomputer Applications* 1992; **6**(1):50–68.
7. Kirby M, Sirovich L. Application of the Karhunen–Loève procedure for the characterization of human faces. *IEEE Transactions of Pattern Analysis and Machine Intelligence* 1990; **12**(1):103–108.
8. Newman AJ. Model reduction via the Karhunen–Loève expansion. Part I: an exposition. *Technical Report T.R.96-32*, Institute for Systems Research, 1996.
9. Aubry N, Guyonnet R, Lima R. Spatio-temporal symmetries and bifurcations via bi-orthogonal decompositions. *Journal of Nonlinear Sciences* 1992; **2**:183–215.
10. Aubry N, Guyonnet R, Lima R. Turbulence spectra. *Journal of Statistical Physics* 1992; **67**:203–228.
11. Fletcher CAJ. *Computational Galerkin Methods*. Springer Series in Computational Physics. Springer: New York, 1984.
12. Rempfer D, Fasel HF. Dynamics of three-dimensional coherent structures in a flat-plate boundary layer. *Journal of Fluid Mechanics* 1994; **275**:257–285.
13. Rempfer D. On low-dimensional Galerkin models for fluid flow. *Theoretical and Computational Fluid Dynamics* 2000; **14**(2):75–88.
14. Noack BR, Eckelmann H. A low-dimensional Galerkin method for the three-dimensional flow around a circular cylinder. *Physics of Fluids* 1994; **6**:124–143.

15. Noack BR, Afanasiev K, Morzynski M, Thiele F. A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. *Journal of Fluid Mechanics* 2003; **497**:335–363.
16. Noack BR, Papas P, Monkewitz PA. The need for a pressure-term representation in empirical Galerkin models of incompressible shear flows. *Journal of Fluid Mechanics* 2005; **523**:339–365.
17. Sirisup S, Karniadakis GE. A spectral viscosity method for correcting the long-term behavior of POD models. *Journal of Computational Physics* 2004; **194**(1):92–116.
18. Perret L, Collin E, Delville J. Polynomial identification of POD based low-order dynamical system. *Journal of Turbulence* 2006; **7**(1):1–15.
19. Galletti B, Bottao A, Bruneau C-H, Iollo A. Accurate model reduction of transient and forced wakes. *European Journal of Mechanics B/Fluids* 2007; **26**:354–366.
20. Couplet M, Basdevant C, Sagaut P. Calibrated reduced order POD Galerkin system for fluid flow modeling. *Journal of Computational Physics* 2005; **207**:192–220.
21. Bard Y. *Nonlinear Parameter Estimation*. Academic Press: New York, 1974.
22. Benson M. Parameter fitting in dynamic models. *Ecological Modelling* 1979; **6**:97–115.
23. Carlin B, Polson N, Stoffer D. A Monte-Carlo approach to nonnormal and nonlinear state space modeling. *Journal of the American Statistical Association* 1992; **87**:493–500.
24. Bremer C, Kaplan D. Markov chain Monte-Carlo estimation of nonlinear dynamics from time series. *Physica D* 2001; **160**:116–126.
25. Wang G. Parameter optimization in large scale dynamical systems: a method of contractive mapping. *Mathematics and Computers in Simulation* 2004; **66**:565–576.
26. Müller T, Timmer J. Fitting parameters in partial differential equations from partially observed noisy data. *Physica D* 2002; **171**:1–7.
27. Baake E, Baake M, Bock H, Briggs K. Fitting ordinary differential equations to chaotic data. *Physical Review A* 1992; **45**:5524–5529.
28. Bock H. Recent advances in parameter identification for ordinary differential equations. *Progress in Scientific Computing* 1983; **2**:95–121.
29. Voss H, Bünner M, Abel M. Identification of continuous spatiotemporal systems. *Physical Review E* 1998; **57**:2820–2823.
30. Varah J. A spline least-squares method for numerical parameter estimation in differential equations. *SIAM Journal on Scientific and Statistical Computing* 1982; **3**(1):28–46.
31. Ramsay J, Hooker G, Campbell D, Cao J. Parameter estimation of differential equations: a generalized smoothing approach. *Journal of the Royal Statistical Society*, *Series B* 2007; **69**(5):741–796.
32. Poyton A, Varziri M, McAuley K, McLellan P, Ramsay J. Parameter estimation in continuous time dynamic models using principal differential analysis. *Computers and Chemical Engineering* 2006; **30**:698–708.
33. Ramsay J. Principal differential analysis: data reduction by differential equations. *Journal of the Royal Statistical Society*, *Series B* 1996; **58**(3):495–508.
34. Shepard D. A two dimensional interpolation function for irregularly spaced data. *Proceedings of 23rd National Conference ACM*, New York, 1968; 517–524.
35. Hardy R. Multiquadric equations of topography and other irregular surfaces. *Journal of Geophysical Research* 1971; **76**:1905–1915.
36. Fasshauer GE. *Meshfree Approximation Methods with MATLAB*. Interdisciplinary Mathematical Sciences, vol. 6. World Scientific Publisher: Singapore, 2007.
37. Kansa E. 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* 1990; **19**(8/9):147–161.
38. Zerroukat T, Power H, Chen C. A numerical method for heat transfer problems using collocation and radial basis functions. *International Journal of Numerical Methods in Engineering* 1998; **42**:1263–1278.
39. Mai-Duy N, Tran-Cong T. Numerical solution of Navier–Stokes equations using multiquadric radial basis function networks. *International Journal of Numerical Methods in Fluids* 2001; **37**:65–86.
40. Fasshauer GE. Solving differential equations with radial basis functions: multilevel methods and smoothing. *Advances in Computational Mathematics* 1999; **11**:139–159.
41. Hoerl AE, Kennard RW. Ridge regression: biased estimation for nonorthogonal problems. *Technometrics* 1970; **12**(3):55–76.
42. Cristianini N, Taylor JS. *An Introduction to Support Vector Machines and other Kernel-based Learning Methods*. Cambridge University Press: Cambridge, 2000.

43. Rosipal R, Trejo L. Kernel partial least-squares regression in reproducing Kernel Hilbert space. *Journal of Machine Learning Research* 2001; **2**:97–123.
44. Kitanidis P. *Introduction to Geostatistics. An Application to Hydrogeology*. Cambridge University Press: U.K., 1997.
45. Chng ES, Chen S, Mulgrew B. Gradient radial basis function networks for nonlinear and nonstationary time series prediction. *IEEE Transactions on Neural Networks* 1996; **7**(1):190–194.
46. Casdagli M. Nonlinear prediction of chaotic time series. *Physica D* 1989; **35**:335–356.
47. Casdagli M. Chaos and deterministic versus stochastic nonlinear modeling. *Journal of the Royal Statistical Society*, *Series B* 1992; **54**:303–328.
48. Broomhead DS, Lowe D. Multivariable functional interpolation and adaptive networks. *Complex Systems* 1988; **2**:321–355.
49. Potts MAS, Broomhead DS. Time series prediction with a radial basis function network. *SPIE Adaptive Signal Processing* 1991; **1565**:225–266.
50. Cowper M, Mulgrew B, Unsworth C. Nonlinear prediction of chaotic signals using a normalized radial basis function network. *Signal Processing* 2002; **82**:775–789.
51. Ni X, Simons S. Nonlinear dynamic system identification using radial basis function network. *Proceedings of 35th Conference on Decision and Control*, Kobe, Japan, 1996.
52. Mann I, McLaughlin S. Dynamical system modelling using radial basis functions. *Adaptive Systems for Signal Processing*, *Communications*, *and Control Symposium*, Lake Louise, Alta., Canada, 2000; 461–465.
53. Kincaid D, Cheney W. *Numerical Analysis*: *Mathematics of Scientific Computing* (3rd edn). Brooks/Cole: Pacific Grove, CA, 2002.
54. Lorang L, Abéguilé F, Fraigneau Y, Tenaud C. Identification de systémes dynamiques dans un canal plan turbulent á laide de réseaux de neurones. *18éme Congrés Français de Mécanique*, Grenoble, France, 2007.
55. Noack B, Schlegel M, Ahlborn B, Mutschke G, Morzynski M, Comte P, Tadmor G. A finite-time thermodynamics formalism for unsteady flows. *Journal of Non-Equilibrium Thermodynamics* 2008; **33**:103–148.
56. Rippa S. An algorithm for selecting a good value for the parameter '*c*' in the radial basis function interpolation. *Advances in Computational Mathematics* 1999; **11**:193–210.
57. Fasshauer GE, Zhang J. On choosing optimal shaping parameters for RBF approximations. *Numerical Algorithms* 2007; **45**:345–368.
58. Reiterer P, Lainscsek C, Schurrer F, Letellier C, Maquet J. A nine-dimensional Lorenz system to study high-dimensional chaos. *Journal of Physics A*: *Mathematical and General* 1998; **31**:7121–7139.
59. Kassam A, Trefethen L. Fourth order time stepping for stiff PDE's. *SIAM Journal on Scientific Computing* 2005; **26**:1214–1233.
60. Wolf A, Swift J, Swinnet H, Vastano J. Determining Lyapunov exponents from a time series. *Physica D* 1985; **16**:285.
61. Shimada I, Nagashima T. A numerical approach to ergodic problems on dissipative dynamical systems. *Progress of Theoretical Physics* 1979; **61**:1605–1616.
62. Benettin G, Galgani L, Giorgilli A, Strelcyn J. Lyapunov characteristic exponents for smooth dynamical systems and for Hamiltonian systems a method for computing all of them. *Meccanica* 1980; **15**:9.