A Versatile Nonlinear Method for Predictive Modeling

Meng-Sing Liou[1] and Weigang Yao[2]
NASA Glenn Research Center, Cleveland, OH 44135, USA

As computational fluid dynamics techniques and tools become widely accepted for real-world practice today, it is intriguing to ask: what areas can it be utilized to its potential in the future. Some promising areas include design optimization and exploration of fluid dynamics phenomena (the concept of numerical wind tunnel), in which both have the common feature where some parameters are varied repeatedly and the computation can be costly. We are especially interested in the need for an accurate and efficient approach for handling these applications: (1) capturing complex nonlinear dynamics inherent in a system under consideration and (2) versatility (robustness) to encompass a range of parametric variations. In our previous paper, we proposed to use first-order Taylor expansion collected at numerous sampling points along a trajectory and assembled together via nonlinear weighting functions. The validity and performance of this approach was demonstrated for a number of problems with a vastly different input functions. In this study, we are especially interested in enhancing the method's accuracy; we extend it to include the second-orer Taylor expansion, which however requires a complicated evaluation of Hessian matrices for a system of equations, like in fluid dynamics. We propose a method to avoid these Hessian matrices, while maintaining the accuracy. Results based on the method are presented to confirm its validity.

I. Introduction

TO make a CFD solution practically and broadly useful, it must be fast, easy to set up, robust and accurate. Examples such as design optimization and exploration of flow scenarios with different inputs/parameters require repetitive calculations of solutions varying over the parameters space. Hence, it is enormously important to have a *predictive* model that possesses the following property: (1) accuracy, (2) efficiency, and (3) robust applicability. The first property demands that the model retains the accuracy of the full solution (whether it be Euler or Navier-Stokes equations based); the second property is to require only a fraction of the cost of the full solution; and the third property is to require that the model be applicable over a sufficiently large variation of parameters, even revealing some physical phenomena not seen in the baseline solution.

The goal of the study is to take the traditional CFD goal to a level higher—beyond computing a solution corresponding to each set of parameters each time, but producing a model that can be used over different sets of parameters, varying over a sufficiently large space, many times. Simply put, the modeling is based on a given baseline trajectory (configuration), then the model so developed should be able to perform outside of this baseline, in a new configuration substantially different from the baseline.

A key element for the modeling to be successful is to be able to preserve *nonlinearity* that exists in problems of interest, thus, requiring at least a nonlinear formulation in the modeling from the outset. The modeling principle should be applicable to a range of governing equations; for fluid dynamics they can be Euler or Navier-Stokes (including RANS, LES, and DNS) equations.

The objective of this paper is to present the concept of our nonlinear modeling approach; we shall describe the basic concept and steps for constructing such a model and then demonstrate its efficacy of the above three properties for problems of practical interest in aerodynamics. Cases included in this paper are based on the Euler equations since our interest primarily lies in unsteady flow phenomena in which the main physical driving force is the inviscid one. However, the same approach can be applied to a steady flow problem where the input parameter can be a new stationary Mach number or a new flight angle of attack.

---

[1] Senior Technologist, Propulsion Division, 21000 Brookpark Road, MS 5-11, Cleveland, OH 44135. Fellow.
[2] Postdoc Fellow, NASA Postdoc Program. Currently Research Fellow Associate, School of Mechanical and Aerospace Engineering, Queen`s University, Belfast, UK. Member.

Let us consider a time-dependent nonlinear system,

$$\frac{\partial \mathbf{x}}{\partial t} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{1}$$

$$\mathbf{x} = [x_1, x_2, \cdots, x_n]^T, \quad \mathbf{f} = [f_1, f_2, \cdots, f_n]^T, \quad \mathbf{u} = [u_1, u_2, \cdots, u_m]^T \tag{2}$$

where $\mathbf{x}$ is a $n\times1$ state vector and $\mathbf{u}$ a $m\times1$ input vector. A common approach for constructing a nonlinear model to meet the above three properties for accuracy, efficiency, and robustness (versatility) is to formulate the problem in a reduced space in order to reduce the computation effort. An appropriate projection can be chosen to construct a nonlinear ROM in a reduced basis ϕ, expressed as

$$\dot{\mathbf{x}}_r = \phi^T \mathbf{f}(\phi \mathbf{x}_r, \mathbf{u}) \tag{3}$$

where $\mathbf{x}_r$ is a $r\times1$ state vector, with $r << n$. In Eq.(3), it is critically important to have an accurate nonlinear full-order model $\mathbf{x}$ before applying projection; a proper construct of the projection matrix ϕ is a key step that affects the success of the modeling. Rewienski and White[1] introduced the trajectory piecewise-linear (TPWL) approach for constructing a nonlinear model: including the projections of state variables and the governing nonlinear differential equations onto a subspace together with a proper weighting of a number of truncated linear models that are expanded about some operating points (or we preferably call sampling points in this paper). Gu and Roychowdhury[2] proposed a similar trajectory-based approach by projecting the full order model into a low order nonlinear manifold subspace and demonstrated improved accuracy by their method for analog circuits and a biochemical system. Gratton and Willcox[3] combined the TPWL and proper orthogonal decomposition (POD) methods to derive a reduced model to study a flow control scheme for a supersonic diffuser. He et al.[4] pointed out instability arising from applying the TPWL method with the POD method, or to over-fitting problems. They discussed that instability appear when including basis vector that corresponds to smaller eigenvalue in the POD method.

Preserving nonlinearity of the problem under study is utmost important for obtaining a model's predictive capability. The work by Rewienski and White [1] on micromachine devices inspired us to use piecewise linear local solutions for study of nonlinear unsteady aerodynamics. In the context of control theory, gain scheduling is a very common practice to approximate a nonlinear control plant[5], and it shares the same concept with TPWL, that is, divide the complex nonlinear model into sub-models[6]. Computational cost can be reduced by gain scheduling pre-computed sub-models. In this sense, TPWL also fits in a gain scheduling framework, and more specifically a linear parameter varying (LPV) approach[7], which approximates nonlinear system by assembling locally linear models obtained by Jacobian linearization at operation points[8]. Since LPV method depends on local linear model or first order Taylor expansion, therefore, the variation from operation points cannot be very large[9]. Other nonlinear modeling methods have also been proposed, such as the Discrete Empirical Interpolation Method (DEIM) by Chaturantabut and Sorensen[10] and the Gauss-Newton with approximated tensors (GNAT) method by Carlberg et al.[11]. Both DEIM and GNAT apply POD basis for model reduction.

Despite successes of POD reported in the literature, some difficulties and limitations are also present for applications to complicated problems in CFD. And we would approach nonlinear modeling with a different idea guided by the following principal. We shall retain accuracy of modeling (preferably not losing resolution of the original system by reducing the degree of freedom) and achieve efficiency so that the resulting model will be valid and economic for a "reasonably wide" variation of parameter's types and values.

Previously, we presented a nonlinear reduced model via recurrent artificial neural network training[12], by assembling a number of weighted neurons associated with different inputs, in which the weights are based on the radial basis functions[13]. This thinking follows the practice used in metamodeling. The present work utilizes a similar concept for constructing a nonlinear model by *collecting a set of submodels that are each valid in the neighborhood of sampling points.* The idea of TPWL comes handy so that we can simply replace the neurons with the piecewise linear solutions valid around the sampling points. It is noted that our approach does not belong in the realm of model reduction because we do not employ the reduced basis matrix ϕ and the dimension in our approximate system remains the same as the full order model.

In the next section, we shall give a detailed description of the mathematical foundation of our method for nonlinear modeling that seeks to realize the three properties stated in the beginning. Specifically, we show how the submodel at a sampling point is constructed through Taylor series expansion and how these submodels are combined through weighting functions to form the final model. Our previous publication[14] has presented the idea and formulation utilizing the first-order Taylor series and the method is called weighted piecewise linear (WPL) modeling; the mathematics involved is relatively straightforward. The present paper focus on the second-order Taylor expansion and the method is now called weighted piecewise quadratic (WPQ) modeling; the algebra

becomes more complicated resulting from the Hessian matrix and seeking its approximation is the subject of this study.

II. Formulation

In order to determine the evolution of $\mathbf{x}$ in Eq. (1) accurately, it is necessary to be able to evaluate $\mathbf{f}$ accurately, which in general is a complex nonlinear function of $\mathbf{x}$. A straightforward idea is to base on the local information of $\mathbf{f}$ using Taylor's series expansion about $\mathbf{x}_0$, as also standard for integrating differential equations. But the validity of the expansion is limited to a small region of $\mathbf{x}_0$ along the evolution (trajectory) of $\mathbf{x}$. Instead of utilizing just a single state, we shall at will approximate $\mathbf{f}$ with a series of p locally accurate solutions $\mathbf{f}_i, i = 1,2,\cdots,p$ valid around sampling points $\mathbf{x}_{0,i}$ and we can assemble them together by weighting functions w_i.

$$\mathbf{f}(\mathbf{x},\mathbf{u}) \approx \sum_{i=1}^{p} w_i(\mathbf{x},\mathbf{u};\mathbf{x}_{0,i},\mathbf{u}_{0,i})\mathbf{f}_i(\mathbf{x},\mathbf{u};\mathbf{x}_{0,i},\mathbf{u}_{0,i}) \tag{4}$$

where

$$\begin{aligned}\mathbf{f}_i(\mathbf{x},\mathbf{u};\mathbf{x}_{0,i},\mathbf{u}_{0,i}) = \mathbf{f}^{(0)}(\mathbf{x}_{0,i},\mathbf{u}_{0,i}) + \mathbf{f}^{(1)}(\mathbf{x};\mathbf{x}_{0,i},\mathbf{u}_{0,i}) + \mathbf{f}^{(2)}(\mathbf{x};\mathbf{x}_{0,i},\mathbf{u}_{0,i}) + \cdots \\ + \mathbf{g}^{(1)}(\mathbf{u};\mathbf{x}_{0,i},\mathbf{u}_{0,i}) + \mathbf{g}^{(2)}(\mathbf{u};\mathbf{x}_{0,i},\mathbf{u}_{0,i}) + \cdots\end{aligned} \tag{5}$$

with $\mathbf{f}^{(0)}$ denoting the function value and $\mathbf{f}^{(n)}$, n>0, the nth derivative with respect to $\mathbf{x}$, and they all are evaluated at the sampling point ($\mathbf{x}_{0,j}$,$\mathbf{u}_{0,j}$). For example,

$$\begin{aligned}&\mathbf{f}^{(0)}(\mathbf{x}_{0,i},\mathbf{u}_{0,i}) = \mathbf{f}(\mathbf{x}_{0,i},\mathbf{u}_{0,i}) \\ &\mathbf{f}^{(1)}(\mathbf{x};\mathbf{x}_{0,i},\mathbf{u}_{0,i}) = D_x\mathbf{f}(\mathbf{x}_{0,i},\mathbf{u}_{0,i})\cdot\Delta\mathbf{x}, \quad D_x\mathbf{f}(\mathbf{x},\mathbf{u}) = \frac{\partial\mathbf{f}}{\partial\mathbf{x}}, \quad \Delta\mathbf{x} = \mathbf{x} - \mathbf{x}_{0,i} \\ &\mathbf{f}^{(2)}(\mathbf{x};\mathbf{x}_{0,i},\mathbf{u}_{0,i}) = \frac{1}{2}\Delta\mathbf{x}^T\cdot D_x^2\mathbf{f}(\mathbf{x}_{0,i},\mathbf{u}_{0,i})\cdot\Delta\mathbf{x}, \quad D_x^2\mathbf{f}(\mathbf{x},\mathbf{u}) = \frac{\partial^2\mathbf{f}}{\partial\mathbf{x}^2}\end{aligned} \tag{6}$$

where $D_x\mathbf{f}(\mathbf{x},\mathbf{u})$ is the usual Jacobian matrix of $\mathbf{f}$ with respect to $\mathbf{x}$ and $D_x^2\mathbf{f}(\mathbf{x},\mathbf{u})$ is the Hessian matrix. Evaluation of the Hessian matrix for fluid equations is complicated and expensive, which is the main reason for it being scarcely touched in practice. In our earlier work, we also stopped with the first-order expansion. But there is a good reason for including the second-order term, such as for enhancing accuracy and allowing a smaller number of sampling points while keeping same accuracy, and for automatically including nonlinearity in the formulation. It is the primary motivation for the current study in which we seek to develop an effective strategy to approximate the Hessian matrix so that the accuracy of second-order expansion is realized while the computational effort for this additional accuracy is minimized.

Similarly $\mathbf{g}^{(n)}$, n>0, is the nth derivative denoting the derivative of $\mathbf{f}$ with respect to $\mathbf{u}$, also evaluated at the sampling point ($\mathbf{x}_{0,j}$,$\mathbf{u}_{0,j}$),

$$\begin{aligned}&\mathbf{g}^{(1)}(\mathbf{u};\mathbf{x}_{0,i},\mathbf{u}_{0,i}) = D_u\mathbf{f}(\mathbf{x}_{0,i},\mathbf{u}_{0,i})\cdot\Delta\mathbf{u}, \quad D_u\mathbf{f}(\mathbf{x},\mathbf{u}) = \frac{\partial\mathbf{f}}{\partial\mathbf{u}}, \quad \Delta\mathbf{u} = \mathbf{u} - \mathbf{u}_{0,i} \\ &\mathbf{g}^{(2)}(\mathbf{u};\mathbf{x}_{0,i},\mathbf{u}_{0,i}) = \frac{1}{2}\Delta\mathbf{u}^T\cdot D_u^2\mathbf{f}(\mathbf{x}_{0,i},\mathbf{u}_{0,i})\cdot\Delta\mathbf{u}, \quad D_u^2\mathbf{f}(\mathbf{x},\mathbf{u}) = \frac{\partial^2\mathbf{f}}{\partial\mathbf{u}^2}\end{aligned} \tag{7}$$

Finally, the weighting function used in this work is based on the radial basis function (RBF)[13] of the Euclid distance between the state $\mathbf{x}$ and the sample (operating) points $\mathbf{x}_{0,i}$, given as follows:

$$w_i = e^{\frac{-\|\mathbf{x}-\mathbf{x}_{0,i}\|}{\delta_i^2}} \Big/ \sum_{j=1}^{p} e^{\frac{-\|\mathbf{x}-\mathbf{x}_{0,j}\|}{\delta_j^2}} \tag{8}$$

where δ_i is a representative length scale measuring the spread of the solution of the *i*th sampling point. We set δ_i=1/√50 for all calculations in this study; this choice seems to work well broadly for the variety of problems. No attempt has been made to find a better specification of this parameter. It is reminded that the weighting function is normalized so that $\sum_{i=1}^{p} w_i = 1$.

In our previous work, we have included the first-order expansions $\mathbf{f}^{(1)}$ and $\mathbf{g}^{(1)}$, hence called piecewise linear (WPL) in Rewienski and White[1] ; its power and versatility for the Euler equations for several unsteady aerodynamic problems with different input functions are documented[14]. Since this model is nonlinear via the weighting function, we also call it weighted piecewise linear (WPL) modeling, to avoid the impression for a linear model.

In our current work, we begin by including the second-order expansions, hence the new formulation will be appropriately named weighted piecewise quadratic (WPQ) modeling. Including the Hessian matrices $D_x^2\mathbf{f}$ and $D_u^2\mathbf{f}$ however leads to a formulation with algebraic complication and computational expenses for a fluid dynamics systems, despite advantages of increased accuracy and range of trust region. In what follows we shall further explore the Hessian terms and attempt to approximate it so that there is no explicit appearance of it, thus significantly reducing the complexity of the method and computational cost.

For illustration, it is sufficient to consider the vector $\mathbf{f}(\mathbf{x},\mathbf{u})$ component by component, namely taking f_i individually. Thus, we shall take the first component f_1 for illustration. Also, it suffices to consider the Taylor series expansion at an arbitrary sampling point since the formula remains the same for all sampling points. Hence, we shall drop the subscript associated with a specific point. And we shall focus on evaluating $f_1^{(2)}(\mathbf{x};\mathbf{x}_0,\mathbf{u}_0)$, while the same procedure can be applied to $g_1^{(2)}(\mathbf{x};\mathbf{x}_0,\mathbf{u}_0)$. Here we have

$$f_1^{(2)}(\mathbf{x};\mathbf{x}_0,\mathbf{u}_0) = \frac{1}{2}\Delta\mathbf{x}^T D_x^2 f_1(\mathbf{x}_0,\mathbf{u}_0)\Delta\mathbf{x}, \quad \Delta\mathbf{x} = \mathbf{x}-\mathbf{x}_0 \tag{9}$$

where

$$D_x^2 f_1(\mathbf{x},\mathbf{u}) = \begin{bmatrix} f_{1,x_1x_1} & f_{1,x_1x_2} & \cdots & f_{1,x_1x_n} \\ f_{1,x_2x_1} & f_{1,x_2x_2} & \cdots & f_{1,x_2x_n} \\ \vdots & \vdots & \vdots & \vdots \\ f_{1,x_nx_1} & f_{1,x_nx_2} & \cdots & f_{1,x_nx_n} \end{bmatrix}, \quad f_{1,x_ix_j} = \frac{\partial^2 f_1}{\partial x_i \partial x_j} \tag{10}$$

A key observation that renders simplifying the evaluation of the Hessian is explained below. Let us denote

$$\Delta\mathbf{x}^T D_x^2 f_1(\mathbf{x}_0,\mathbf{u}_0) = \left[d_1, d_2, \cdots, d_n\right] \tag{11}$$

where a typical element $d_j(\mathbf{x};\mathbf{x}_0)$ in the above vector is given by

$$\begin{aligned} d_j(\mathbf{x};\mathbf{x}_0,\mathbf{u}_0) &= \sum_{i=1}^{n} \Delta x_i f_{1,ij}(\mathbf{x}_0,\mathbf{u}_0) \\ &= \Delta x_1 \frac{\partial^2 f_1}{\partial x_1 \partial x_j} + \Delta x_2 \frac{\partial^2 f_1}{\partial x_2 \partial x_j} + \cdots + \Delta x_n \frac{\partial^2 f_1}{\partial x_n \partial x_j} \end{aligned} \tag{12}$$

Approximate the Hessian derivatives by forward differencing,

$$
\begin{aligned}
\frac{\partial^2 f_1}{\partial x_1 \partial x_j}\Delta x_1 &= \left.\frac{\partial f_1}{\partial x_j}\right|_{x_1=x_1+\Delta x_1, x_j\,(j\neq 1)} - \left.\frac{\partial f_1}{\partial x_j}\right|_{\mathbf{x}_0} \\
\frac{\partial^2 f_1}{\partial x_2 \partial x_j}\Delta x_2 &= \left.\frac{\partial f_1}{\partial x_j}\right|_{x_2=x_2+\Delta x_2, x_j\,(j\neq 2)} - \left.\frac{\partial f_1}{\partial x_j}\right|_{\mathbf{x}_0} \\
&\vdots \\
\frac{\partial^2 f_1}{\partial x_n \partial x_j}\Delta x_n &= \left.\frac{\partial f_1}{\partial x_j}\right|_{x_n=x_n+\Delta x_n, x_j\,(j\neq n)} - \left.\frac{\partial f_1}{\partial x_j}\right|_{\mathbf{x}_0}
\end{aligned}
\tag{13}
$$

By substitution of the above approximations, d_j becomes

$$
\begin{aligned}
d_j(\mathbf{x};\mathbf{x}_0,\mathbf{u}_0) &= \sum_{i=1}^{n} \Delta x_i f_{1,x_i x_j}(\mathbf{x}_0,\mathbf{u}_0) \\
&= \left[f_{1,x_j}(x_1+\Delta x_1, x_{0,k(\neq 1)}, \mathbf{u}_0) - f_{1,x_j}(\mathbf{x}_0,\mathbf{u}_0) \right] + \left[f_{1,x_j}(x_2+\Delta x_2, x_{0,k(\neq 2)}, \mathbf{u}_0) - f_{1,x_j}(\mathbf{x}_0,\mathbf{u}_0) \right] \\
&\quad + \cdots + \left[f_{1,x_j}(x_n+\Delta x_n, x_{0,k(\neq n)}, \mathbf{u}_0) - f_{1,x_j}(\mathbf{x}_0,\mathbf{u}_0) \right] \\
&= \sum_{i=1}^{n} f_{1,x_j}(x_i+\Delta x_i, x_{0,k(\neq i)}, \mathbf{u}_0) - n f_{1,x_j}(\mathbf{x}_0,\mathbf{u}_0)
\end{aligned}
\tag{14}
$$

Define a mid-point, $\overline{\mathbf{x}}$,

$$
\overline{\mathbf{x}} = \frac{1}{2}(\mathbf{x}+\mathbf{x}_0) = \mathbf{x}_0 + \frac{\Delta \mathbf{x}_0}{2}, \quad \mathbf{x} = \mathbf{x}_0 + \Delta \mathbf{x}_0 \tag{15}
$$

The first term in Eq. (13) is further expanded at this midpoint, it is then contracted to give a considerable simplification,

$$
d_j = n \left.\frac{\partial f_1}{\partial x_j}\right|_{\overline{\mathbf{x}}} - n \left.\frac{\partial f_1}{\partial x_j}\right|_{\mathbf{x}_0} \tag{16}
$$

We call attention to the following: (1) only the first derivatives, no longer second derivatives, are needed, and (2) they are evaluated only at two states, $(\overline{\mathbf{x}}, \mathbf{x}_0)$.

Finally, we have the second-order expansion now in terms of a difference of first derivatives respectively evaluated at the mid-point and base states

$$
\begin{aligned}
f_1^{(2)}(\mathbf{x};\mathbf{x}_0,\mathbf{u}_0) &= \frac{1}{2}\Delta \mathbf{x}^T D_x^2 f_1(\mathbf{x}_0,\mathbf{u}_0)\Delta \mathbf{x} = \frac{1}{2}\sum_{i=1}^{n} d_i \Delta x_i \\
&= \frac{n}{2}\sum_{i=1}^{n}\left[\frac{\partial f_1}{\partial x_i}(\overline{\mathbf{x}},\mathbf{u}_0) - \frac{\partial f_1}{\partial x_i}(\mathbf{x}_0,\mathbf{u}_0) \right]\Delta x_i
\end{aligned}
\tag{17}
$$

Replacing the subscript "1" with "j", we get the expression for the *j*th component

$$
f_j^{(2)}(\mathbf{x};\mathbf{x}_0,\mathbf{u}_0) = \frac{n}{2}\sum_{i=1}^{n}\left[\frac{\partial f_j}{\partial x_i}(\overline{\mathbf{x}},\mathbf{u}_0) - \frac{\partial f_j}{\partial x_i}(\mathbf{x}_0,\mathbf{u}_0) \right]\Delta x_i \tag{18}
$$

Thus, the expression for the full vector $\mathbf{f}^{(2)}(\mathbf{x};\mathbf{x}_0,\mathbf{u}_0)$ is now obtained. Remarkably, these first derivatives are already available when evaluating the first-order Taylor terms $f_j^{(1)}$ or $\mathbf{f}^{(1)}(\mathbf{x};\mathbf{x}_0,\mathbf{u}_0)$. This amounts to using the

same information required for the first-order Taylor expansion, except the Jacobian matrices are now evaluated at the average state, instead of the sample points. Similarly, $g_j^{(2)}(\mathbf{u};\mathbf{x}_0,\mathbf{u}_0)$ can be obtained easily. Thus, the *j*th component is summarized as follows.

$$\begin{aligned} f_j(\mathbf{x};\mathbf{x}_0,\mathbf{u}_0) = f_j(\mathbf{x}_0,\mathbf{u}_0) &+ (1-\frac{n}{2})\sum_{i=1}^{n}\frac{\partial f_j}{\partial x_i}(\mathbf{x}_0,\mathbf{u}_0)\Delta x_i + \frac{n}{2}\sum_{i=1}^{n}\frac{\partial f_j}{\partial x_i}(\bar{\mathbf{x}},\mathbf{u}_0)\Delta x_i \\ &+ (1-\frac{n}{2})\sum_{i=1}^{n}\frac{\partial f_j}{\partial u_i}(\mathbf{x}_0,\mathbf{u}_0)\Delta u_i + \frac{n}{2}\sum_{i=1}^{n}\frac{\partial f_j}{\partial u_i}(\mathbf{x}_0,\bar{\mathbf{u}})\Delta u_i \end{aligned} \tag{19}$$

Now it is instructive to escape the above abstraction and lengthy algebraic manipulation of the *n*-dimensional system and consider a simpler one for bringing some clarity. Let us consider the following two-equation system,

$$\frac{\partial}{\partial t}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f_1(x,y,u) \\ f_2(x,y,u) \end{bmatrix} \tag{20}$$

As before, the local Taylor's series expansions about a sampling point "*i*" $(x_{0,i},y_{0,i},u_{0,i})$ reads

$$\begin{aligned} \mathbf{f}(x,y,u) = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}(x,y,u) \approx \mathbf{f}^{(0)}(x_{0,i},y_{0,i},u_{0,i}) &+ \mathbf{f}^{(1)}(x,y;x_{0,i},y_{0,i},u_{0,i}) + \mathbf{f}^{(2)}(x,y;x_{0,i},y_{0,i},u_{0,i}) \\ &+ \mathbf{g}^{(1)}(u;x_{0,i},y_{0,i},u_{0,i}) + \mathbf{g}^{(2)}(u;x_{0,i},y_{0,i},u_{0,i}) + \cdots \end{aligned} \tag{21}$$

where

$$\mathbf{f}^{(1)}(x,y;x_{0,i},y_{0,i},u_{0,i}) = \begin{bmatrix} f_{1,x} & f_{1,y} \\ f_{2,x} & f_{2,y} \end{bmatrix}_{(x_{0,i},y_{0,i},u_{0,i})}\begin{bmatrix} \Delta x_i \\ \Delta y_i \end{bmatrix}, \qquad \begin{bmatrix} \Delta x_i \\ \Delta y_i \end{bmatrix} = \begin{bmatrix} x - x_{0,i} \\ y - y_{0,i} \end{bmatrix} \tag{22}$$

and

$$\mathbf{f}^{(2)}(x,y;x_{0,i},y_{0,i},u_{0,i}) = \frac{1}{2}\begin{bmatrix} \begin{bmatrix} \Delta x_i & \Delta y_i \end{bmatrix}\begin{bmatrix} f_{1,xx} & f_{1,yx} \\ f_{1,xy} & f_{1,yy} \end{bmatrix}_{(x_{0,i},y_{0,i},u_{0,i})}\begin{bmatrix} \Delta x_i \\ \Delta y_i \end{bmatrix} \\ \begin{bmatrix} \Delta x_i & \Delta y_i \end{bmatrix}\begin{bmatrix} f_{2,xx} & f_{2,yx} \\ f_{2,xy} & f_{2,yy} \end{bmatrix}_{(x_{0,i},y_{0,i},u_{0,i})}\begin{bmatrix} \Delta x_i \\ \Delta y_i \end{bmatrix} \end{bmatrix} \tag{23}$$

with $f_{i,x} = \dfrac{\partial f_i}{\partial x}$, $f_{i,xx} = \dfrac{\partial^2 f_i}{\partial x \partial x}$, $i = 1,2,$ etc.

Similarly,

$$\mathbf{g}^{(1)}(u;x_{0,i},y_{0,i},u_{0,i}) = \begin{bmatrix} f_{1,u} \\ f_{2,u} \end{bmatrix}_{(x_{0,i},y_{0,i},u_{0,i})}\Delta u_i, \qquad \Delta u_i = u - u_{0,i} \tag{24}$$

$$\mathbf{g}^{(2)}(u;x_{0,i},y_{0,i},u_{0,i}) = \frac{1}{2}\begin{bmatrix} f_{1,uu} \\ f_{2,uu} \end{bmatrix}_{(x_{0,i},y_{0,i},u_{0,i})}(\Delta u_i)^2 \tag{25}$$

The second-order Taylor term of f_1 can be expressed as (from Eq. (23))

$$f_1^{(2)}(x,y;x_{0,i},y_{0,i},u_{0,i}) = \frac{1}{2}(d_1\Delta x_i + d_2\Delta y_i) \tag{26}$$

where

$$\begin{aligned} d_1 &= \Delta x_i\, f_{1,xx}\Big|_{(x_{0,i},y_{0,i},u_{0,i})} + \Delta y_i\, f_{1,xy}\Big|_{(x_{0,i},y_{0,i},u_{0,i})} \\ d_2 &= \Delta x_i\, f_{1,yx}\Big|_{(x_{0,i},y_{0,i},u_{0,i})} + \Delta y_i\, f_{1,yy}\Big|_{(x_{0,i},y_{0,i},u_{0,i})} \end{aligned} \tag{27}$$

Applying forward differencing now to approximate the second derivatives, we get

$$\begin{aligned} d_1 &= [f_{1,x}(x;y_{0,i},u_{0,i}) - 2f_{1,x}(x_{0,i},y_{0,i},u_{0,i}) + f_{1,x}(y;x_{0,i},u_{0,i})] \\ d_2 &= [f_{1,y}(x;y_{0,i},u_{0,i}) - 2f_{1,y}(x_{0,i},y_{0,i},u_{0,i}) + f_{1,y}(y;y_{0,i},u_{0,i})] \end{aligned} \tag{28}$$

Expanding the derivatives about an average state, we get the following trapezoidal approximation for d_1 and d_2, as follows.

$$\frac{1}{2}[f_{1,x}(x;y_{0,i},u_{0,i}) + f_{1,x}(y;x_{0,i},u_{0,i})] \approx f_{1,x}(\bar{x},\bar{y},u_{0,i}) \tag{29}$$

and

$$\begin{aligned} d_1 &= [2f_{1,x}(\bar{x},\bar{y},u_{0,i}) - 2f_{1,x}(x_{0,i},y_{0,i},u_{0,i})] \\ d_2 &= [2f_{1,y}(\bar{x},\bar{y},u_{0,i}) - 2f_{1,y}(x_{0,i},y_{0,i},u_{0,i})] \end{aligned} \tag{30}$$

The formulas for $g_1^{(2)}$ and $g_2^{(2)}$ can be obtained similarly; it is even simpler if only one variable *u* is considered in this paper. Putting both the first and second Taylor terms together, we get an approximation of f_1 at the *i*th sampling point $(x_{0,i},y_{0,i},u_{0,i})$

$$f_1(x,y,u) = f_1(x_{0,i},y_{0,i},u_{0,i}) + f_{1,x}(\bar{x},\bar{y},u_{0,i})\Delta x_i + f_{1,y}(\bar{x},\bar{y},u_{0,i})\Delta y_i + f_{1,u}(x_{0,i},y_{0,i},\bar{u})\Delta u_i + \ldots \tag{31}$$

where

$$\bar{x} = \frac{1}{2}(x + x_{0,i}), \quad \bar{y} = \frac{1}{2}(y + y_{0,i}), \quad \bar{u} = \frac{1}{2}(u + u_{0,i}) \tag{32}$$

The formula for f_2 is obtained similarly, by simply replacing the subscript "1" in f_1 with "2".

Then, the complete WPQ for the system Eq. (20) reads

$$\frac{\partial \mathbf{x}}{\partial t} = \mathbf{f}(\mathbf{x},\mathbf{u}) \approx \sum_{i=1}^{p} w_i(\mathbf{x},\mathbf{u};\mathbf{x}_{0,i},\mathbf{u}_{0,i}) \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} (\mathbf{x},\mathbf{u};\mathbf{x}_{0,i},\mathbf{u}_{0,i}) \tag{33}$$

where $\mathbf{x} = (x,y)^T$ and $\mathbf{u}$ is a scalar function if the formula in Eq. (31) is to be used directly. This is all what the WPQ modeling is about and it is ready for a usual time integration provided an input data is given. This is the basis of modeling where a baseline (arbitrary) input function is employed and the integrated is performed; the process is called training. The important piece that shows its power—main impetus of our work, lies in the application of this modeling (training) to a different kind of input functions *without* having to recalculate the original $\mathbf{f}$ for each new input, which is an expensive proposition.

III. Computational Results

We will show in this section the results from using the modeling method detailed above, for problems involving a scalar equation, a system of two equations, and Euler equations with an input that differs vastly in form and magnitude from the baseline. The purpose is to show the power and applicability of the method, with regards to its accuracy, cost savings, and versatility/robustness. As a benchmark solution for comparison for its validity,

integration of the original system will be treated as the "exact" (albeit numerical) solution, which is obtained by using a four-stage Runge-Kutta method.

A. Scalar Equation

We employ a nonlinear scalar equation with an input function u to facilitate understanding of the WPQ modeling.

$$\frac{\partial x}{\partial t} = f(x(t),u(t)) = 5\sin x + x + u \tag{34}$$

The WPQ modeling solves

$$\frac{\partial x}{\partial t} = \sum_{i=1}^{p} w_i [f(x_i,u_i) + a_i(x-x_i) + b_i(u-u_i) + h_i(x-x_i)^2]$$
$$a_i = 5\cos x_i + 1, \quad b_i = 1, \quad h_i = -2.5\sin x_i \tag{35}$$

Here the exact Hessian is used in Eq. (35) as it is readily available; its results also serve as bases for comparison with that of approximate Hessian described above. Specifically, the approximate Hessian developed in previous section gives

$$\frac{\partial x}{\partial t} = \sum_{i=1}^{p} w_i [f(x_i,u_i) + \bar{a}_i(x-x_i) + b_i(u-u_i)]$$
$$\bar{a}_i = 5\cos \bar{x}_i + 1, \quad \bar{x}_i = (x+x_i)/2 \tag{36}$$

For illustration, we arbitrarily choose the following input for training

$$u(t) = \sin 6t + \cos 6t \tag{37}$$

Then the original equation, Eq. (34), is integrated over a specified interval, say $(0,T=2\pi)$, providing a solution at N discrete point (state) and the coefficients, (a_i, b_i, h_i) and $(\bar{a}_i, b_i)$. Out of these N points, we pick p sampling points with their coefficients. The sampling may be arbitrary, we chose an even distribution of points in $(0,T)$; other sampling strategy and its effect are presented in our previous study[14]. These trained coefficients now can be used in Eq. (35) or (36) with the specified sampling points and the integration can proceed as usual to give the approximate solution.

First, we shall use the exact Hessian, Eq. (35). Figure 1 shows the solutions using various sampling points and the comparison with the "exact" solution. The solution with only one sampling point explodes quickly, but with additional sampling points it converges to the exact solution well.

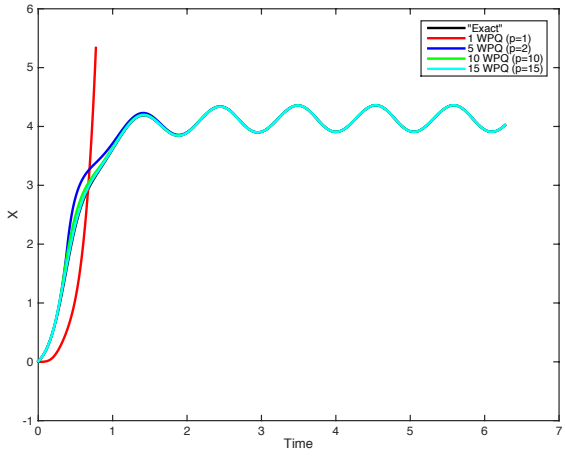


Figure 1. Comparison of various WPQ solutions with the "exact" solution.

Next we use the approximate Hessian and the solution is shown in Figure 2, here the solution with even just one sampling point surprisingly follows the correct trend (compare with Figure 1), the approximate Hessian gives a

faster convergence to the exact solution. In Table 1, one can clearly see a drastic improvement in accuracy by including the second-order expansion term, and the approximate Hessian yields more accurate solutions than the exact Hessian. Moreover, all three methods shows convergency as more submodels (sampling points) are used, i.e., as p is increased. The error is measured, throughout this work, by the following formula.

$$Error = \frac{\frac{1}{N}\sum_{i=1}^{N}\left|\hat{x}_i - x_i\right|}{x_{\max} - x_{\min}}, \quad \hat{x}: \text{exact solution} \tag{38}$$

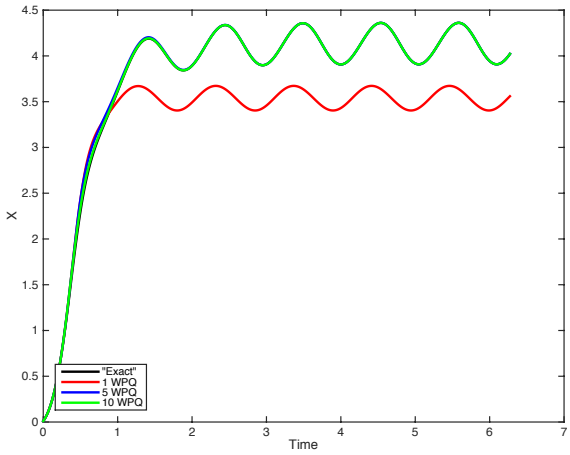


Figure 2. Solutions by the approximate Hessian using different numbers of WPQ submodels.

Table 1. Comparison between first-order and second-order Taylor expansions.

PW Elements	First-order	Exact Hessian	Approximate Hessian
1	NaN	NaN	8.37e-2
2	2.27e-2	1.92e-2	6.15e-3
5	1.20e-2	1.78e-2	5.50e-3
10	8.05e-3	6.82e-3	1.67e-3
20	4.19e-3	8.21e-4	1.74e-4
50	1.15e-3	5.15e-5	1.13e-5
100	3.91e-4	7.16e-6	1.61e-6
150	2.52e-4	2.95e-6	6.87e-7
300	1.70e-4	1.28e-6	3.12e-7

Next, we use the training data obtained with Eq. (37) and extend it to a new input function,

$$u(t) = 5\sin^3 6t \tag{39}$$

The result with 10 WPQ submodels gives an excellent comparison with the exact solution, as shown in Figure 3; the training solution is also included for reference, showing a significant difference between these two solutions, thus confirming the versatility of the approach.

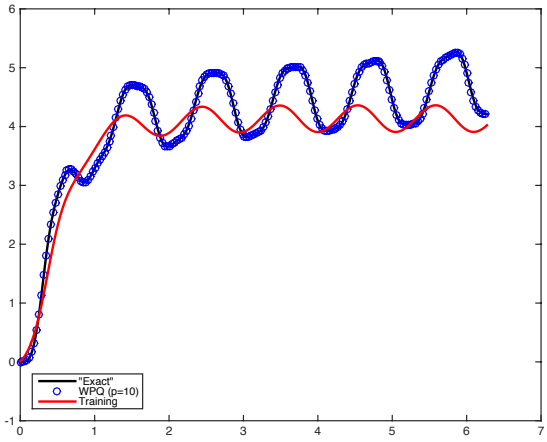


Figure 3. Results of using 10 WPQ submodels for input Eq. (38).

Next, we consider a more complicated input function, which can produce an exploding increase in solution *x*.

$$u(t) = t^2 e^{-\mu t} \sin^3 6t \tag{40}$$

If μ is small, a sudden increase in the solution can occur, as seen in Figure 4 for μ=0.25. The solutions by the exact and approximate Hessians are again included for comparison, in which 1 WPQ again already shows main dynamics of the solution and the 5 WPQ solution essentially coincides with that of 10 WPQ. Noticing again the significant effect of the new input function, compared to the baseline solution. This example shows the reliability of the method for a case with an increasing amplitude in the solution.

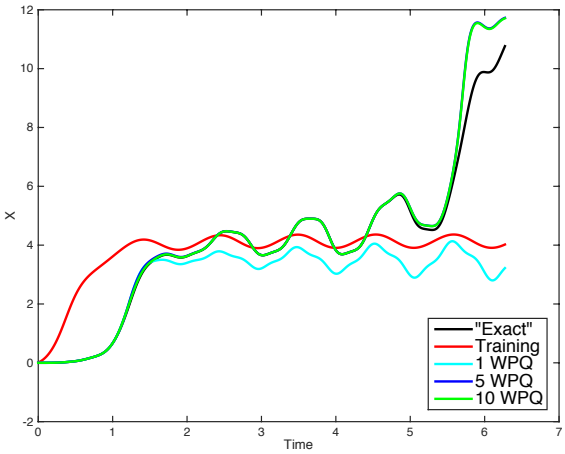


Figure 4. Solutions by the approximate Hessian using different numbers of WPQ submodels, the training and exact solutions are included for comparison.

B. System of Two Equations

Here we specify the 2-equation system considered previously

$$\frac{\partial}{\partial t}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f_1(x,y,u) \\ f_2(x,y,u) \end{bmatrix} \tag{41}$$

with

$$f_1(x,y,u) = \sin(x+y) + u$$
$$f_2(x,y,u) = \sin(xy) - u \tag{42}$$

The input function for training is same as Eq. (37).

Using the trained data, we now apply the model to a new input

$$u(t) = \sin^3 6t \tag{43}$$

The solutions by the first-order and second-order Taylor expansions with 10 submodels are compared with the exact solution for the input Eq. (43) in Figure 5; the first-order solution has a significant discrepancy (even though giving the correct trend) in the x component from the exact solution, but the second-order expansion provides excellent accuracy for both x and y. We note that the first-order solution does converge to the correct solution, by using more submodels, as shown in the same figure.

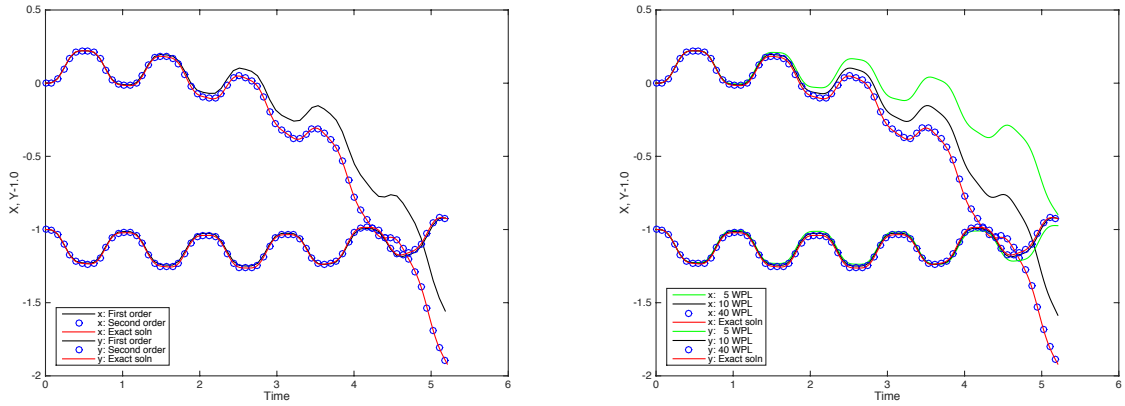


Figure 5. Left: comparison of solutions by the first-order Taylor (WPL) and second-order Taylor (WPQ) expansions with the exact solution, using 10 submodels. Right: convergence of WPL using 5, 10 and 40 submodels.

Using yet another input given in Eq. (40) for the same training data, we get a vastly different result, as shown in Figure 6, with training data included for reference (please note that the vertical coordinate is adjusted in both plots for clarity). In the input with exponential function, the solution again shows increasing amplitudes in both x and y components; the 10 WPQ model gives an excellent comparison with the exact solution, even for cases far from the baseline.

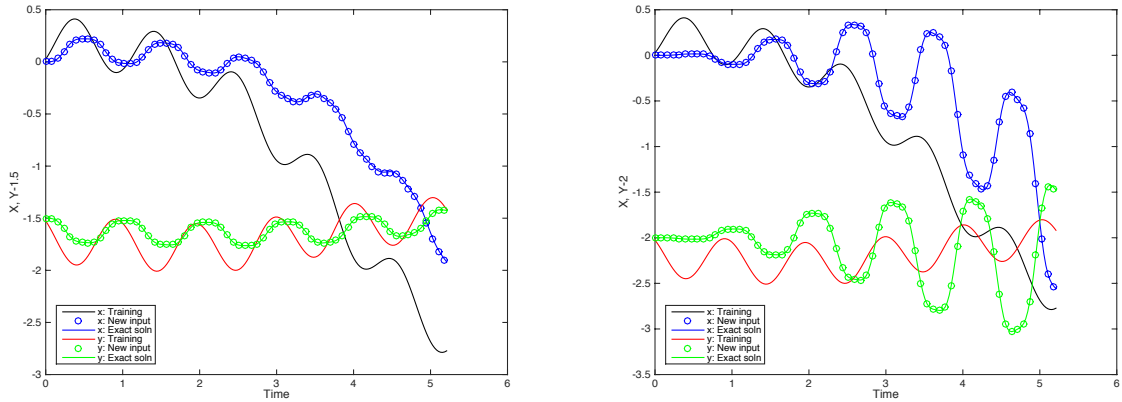


Figure 6. Solutions using 10 WPQ submodels with new inputs, with the training solution (black and red curves) included for comparison. Left: input Eq. (43) and Right: input Eq. (40).

C. Euler Equations

Let us consider the 2D Euler Equations written as:

$$\frac{\partial(\upsilon \mathbf{q})}{\partial t} = \mathbf{R}(\mathbf{q}, \mathbf{u}, \mathbf{v}) \tag{44}$$

where υ is the volume of a computational cell, **q** consists of the conservative flow variables, **R(q,u,v)** is the residual representing spatial derivatives of flow flux functions for a set of input conditions, **u** and **v,** which may be the structural displacement and velocity, such as in a pitching/plunging motion of an airfoil or the flow condition such as the Mach number. The spatial discretization contained in the residual term **R(q,u,v)** can be determined by a state-of-the-art numerical flux method; here we employ the AUSM$^+$-up scheme[15] to approximate the flux function and the van Albada limiter in the MUSCL interpolation of primitive variables.

At the time of finishing up this paper, we have not been able to get the WPQ implemented in the code to work properly. Hence we shall show in the following some typical results presented in the past publication[14], which were obtained by using the weighted piecewise linear (WPL) model, in order to illustrate the efficacy of the approach for a fluid dynamics system. The interested reader is encouraged to refer to our previous paper for details. The WPL model is expressed by

$$\frac{\partial(\upsilon \mathbf{q})}{\partial t} = \sum_{i=1}^{p} w_i(\mathbf{Q}, \mathbf{Q}_i)\{\mathbf{R}(\mathbf{Q}_i) + \mathbf{A}_i(\mathbf{q} - \mathbf{q}_i) + \mathbf{B}_{1,i}(\mathbf{u} - \mathbf{u}_i) + \mathbf{B}_{2,i}(\mathbf{v} - \mathbf{v}_i)\} \tag{45}$$

where the Jacobian matrices are

$$\mathbf{A}_i = \left.\frac{\partial \mathbf{R}}{\partial \mathbf{q}}\right|_{\mathbf{Q}_i}, \quad \mathbf{B}_{1,i} = \left.\frac{\partial \mathbf{R}}{\partial \mathbf{u}}\right|_{\mathbf{Q}_i}, \quad \mathbf{B}_{2,i} = \left.\frac{\partial \mathbf{R}}{\partial \mathbf{v}}\right|_{\mathbf{Q}_i}, \quad \mathbf{Q} = [\mathbf{q}, \mathbf{u}, \mathbf{v}]^T \tag{46}$$

The right-hand term of Eq. (45) gives a weighted combination of p locally linearized models. The evolution of this system is now completely a function of the baseline (training) solutions at the sampling points; it is computationally tractable, requiring only the matrix-vector multiplications and the evaluation of the weights. The evolution of **q** now completely bypasses flux evaluations of the entire flow domain as needed in the full Euler equations for each time step.

We shall see some unsteady aerodynamic characteristics associated with some standard test cases in the AGARD report[14], specifically the so-called "CT2" and "CT5" cases. These two cases have different flight Mach numbers and pitching parameters, as given in Table 2. The pitching motion of the airfoil is described by its angle of attack varying in time from the mean value α_m:

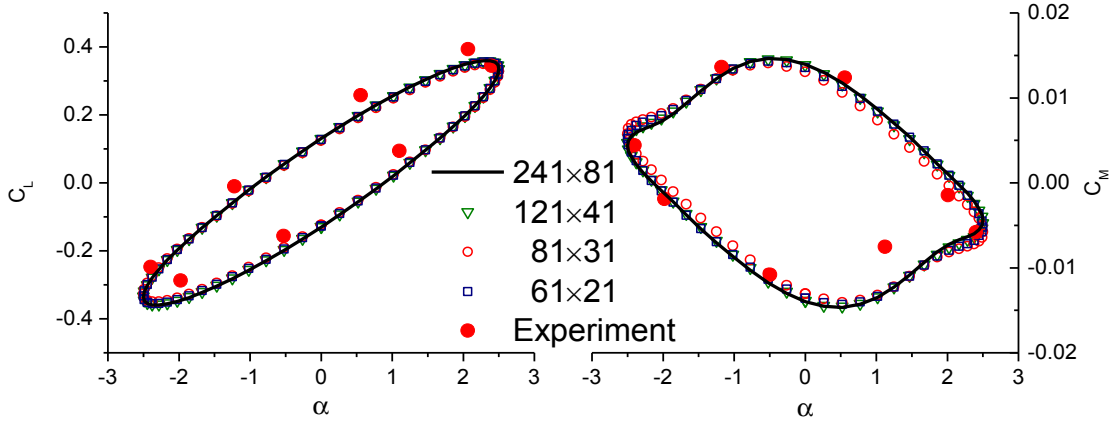
$$\alpha(t) = \alpha_m + \alpha_0 \sin \omega t = \alpha_m + \alpha_0 \sin 2k\tilde{t}, \qquad k = \frac{\omega c}{2U_\infty}, \quad \tilde{t} = \frac{U_\infty t}{c} \tag{47}$$

where α_0 is the oscillation amplitude, k is the reduced frequency defined by the free-stream velocity U_∞ and the chord length c. The pitching motion is centered at x_m.

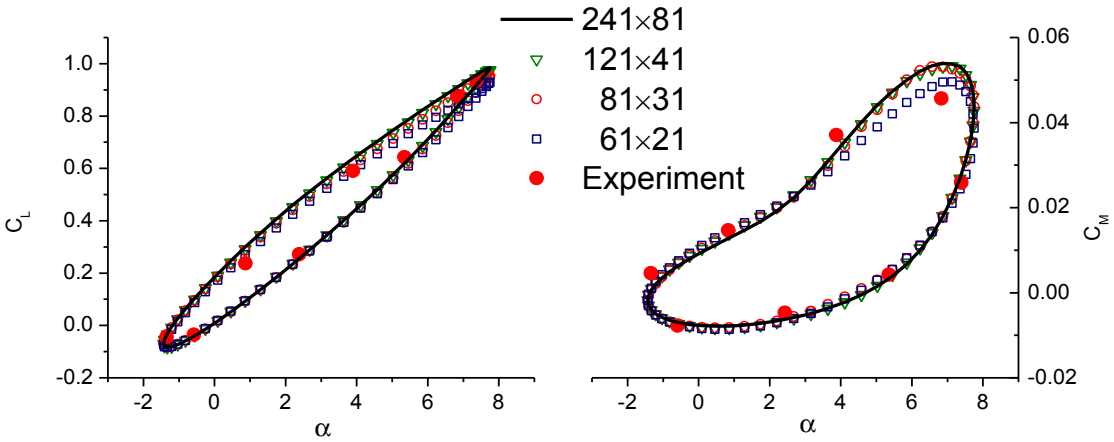
Table 2. Parameters of CT2 and CT5 cases

Case	M_∞	α_m	α_0	k	x_m
CT2	0.6	3.16°	4.59	0.0811	0.273
CT5	0.755	0.016°	2.51	0.0814	0.25

First, we validated the accuracy of the CFD procedure against the experimental data[14]; the comparison of aerodynamic forces, in terms of lift and moment coefficients, in a cycle is given in Figure 7 for both the CT5 and CT2 cases with a sequence of grid refinements, showing the CFD results in good agreement with the data. More importantly, both test cases exhibit a strong nonlinear behavior in the moment coefficient and nonsymmetrical forces between downward and upward motions. It is noted that CT2 has a significantly larger excursion of about 10° in pitching than CT5 with 5°, thus it is expected to show stronger nonlinear effects. In contrast, CT5 operates at a higher Mach number. Both have a varying supersonic pocket in the flowfield during pitching, also manifesting nonlinear effects.



(a) CT5



(b) CT2

Figure 7. Validation of CFD solutions for the AGARD CT5 and CT2 problems.

In what follows, we separately discuss the results from the nonlinear modeling for the CT5 and CT2 problems, as they exhibit rather different flow characteristics even subject to the same input parameters.

C1. CT5

After the training at the baseline condition $\alpha_0 = 2.51°$, $k = 0.0814$, the so-built WPL model can be used for other conditions. Shown in Figure 8 are the results of the full CFD model and the nonlinear model using 31, 21, and 11 WPL solutions with $\alpha_0 = 3.5°$, $k = 0.0814$; it shows that both 21 and 11 WPL models can also faithfully capture nonlinear features of the original CFD system. In contrast, the linear results, obtained by retaining only the zeroth-order Taylor expansion around the steady operation point, completely miss the nonlinear phenomenon in the original system. As the linear model does not contain any time varying information in the model other than via input function, it is clear that collecting time-varying submodels into a complete model is essential and is a key to maintaining accuracy of the model. After the training, the so-built WPL-based model can be used for other conditions.

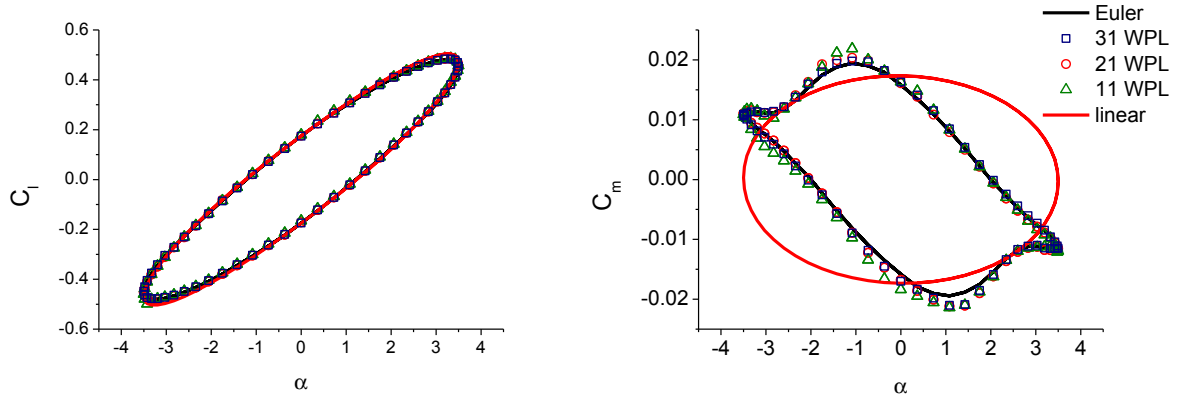


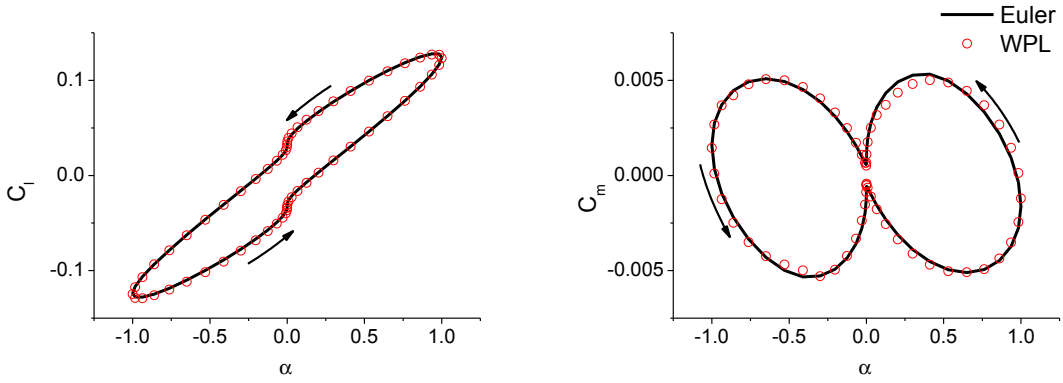
Figure 8. WPL results computed for a different pitching motion $\alpha_0 = 3.5°$, $k = 0.0814$ from the baseline (training) data of $\alpha_0 = 2.51°$ and $k = 0.0814$.

Now we consider the scenario of having a different flight motion from the baseline. In this case we impose a new input signal but with a *cubic* sinusoidal function

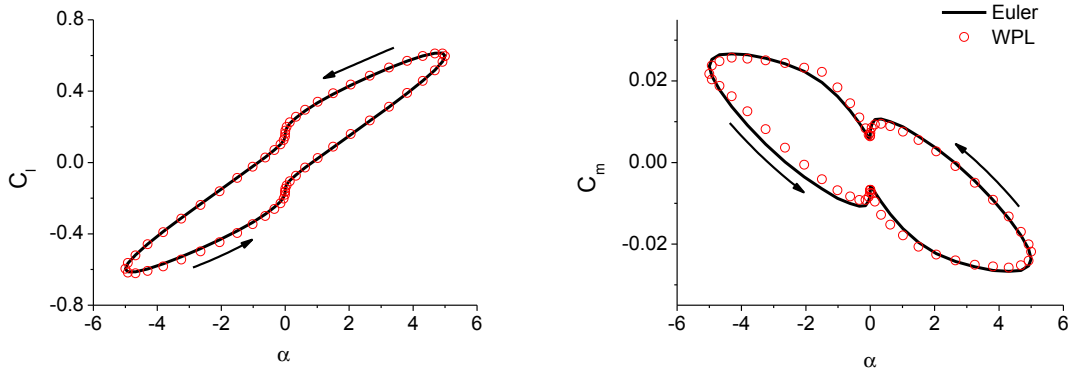
$$\alpha = \alpha_m + \alpha_0 \sin^3(2k\tilde{t}) \tag{48}$$

with the reduced frequency k and α_m unchanged, but with different amplitudes α_0=3.5°, 1.0°and 5.0°. It is noted that the cubic sinusoidal input is expected to give rise to a more complicated flight motion. Also, the amplitude of 5.0° is significantly larger than the baseline value of 2.51°.

Figure 9 shows that the WPL-based model still hold its accuracy even the flight motion is both qualitatively (cubic function) and quantitatively (amplitude) different from the training motion.



(a) $\alpha_0 = 1.0°$



(b) $\alpha_0 = 5.0°$

Figure 9, WPL results with Eq. (48) for amplitude $\alpha_0 = 1.0°$ (a) and and 5.0° (b) departing from the baseline $\alpha_0 = 2.51°$

The proposed modeling, as mentioned earlier, maintains the same degrees of freedom as the full model, unlike the model reduction methods, it is interesting to compare the detailed flowfield predicted by the model with the solution calculated by integrating the full Euler equations. Figure 10 shows pressure contours of both solutions side by side; they are nearly indistinguishable.

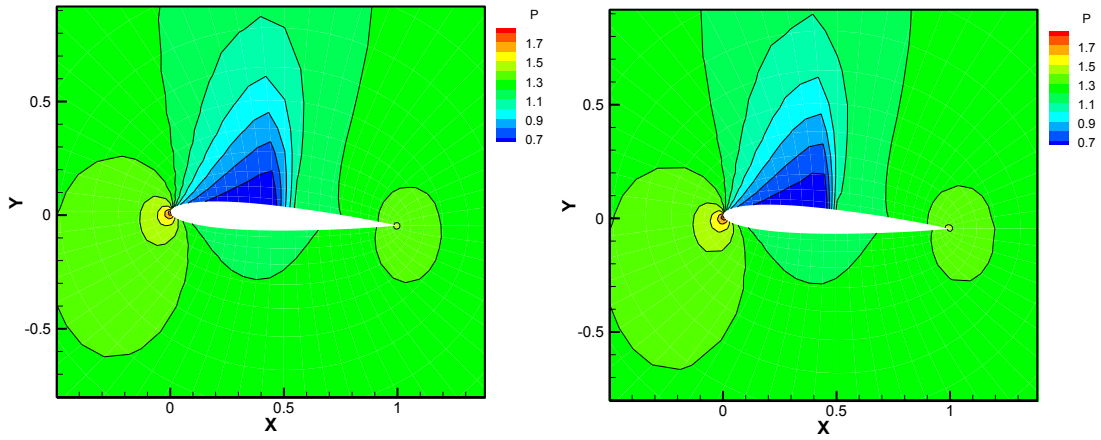


Figure 10. Flowfield expressed by pressure contours at a time slice (t=T/4) with a cosine signal ($\alpha_0 = 3.5°$, Δt = T/60) (Left: Euler, Right: WPL).

Finally, we compare in Table 3 the computational cost for performing a full-order CFD analysis and the additional work required to construct the WPL-based nonlinear model. One should note that this extra overhead is one time only; any computation for other conditions will require a minimal additional effort to perform matrix-vector multiplications and time integrations—only 1/50 (for the current setup) of the cost incurred by a full CFD solution is needed for another new condition by the model. Hence, in this sense, the so-constructed nonlinear model is not only accurate but also computationally economical for parametric studies and design optimization. The more calculations performed, the more savings realized.

Table 3. Comparison of CPU times by full model, WPL model construction, and running a constructed WPL model on an Intel i7 CPU.

Model Choice	CPU Time (min)
Full Model (Euler)	15.03/period
WPL Model Construction	8.533 (for 60 WPL samples/period)
WPL Model	0.324/period

C2. CT2

For another pitching airfoil problem, CT2 [14], we further apply the previously established WPL-based nonlinear model. This problem is more complex aerodynamically than CT5 in the sense that the aerodynamic forces display stronger nonlinearity, as seen in Figure 11, where the lift force has a noticeable nonlinear behavior near high angles of attack, exhibiting asymmetry with respect to angles of attack.

In Figure 11, we show the performance of the WPL-based solutions against the full Euler solutions of different pitching amplitudes. At a small amplitude ($\alpha_0 = 1°$), the flow behaves linearly and the linear model is close to the Euler solution. However, when the amplitude is moderate, nonlinearity first appears in the moment coefficient; then

the lift coefficient exhibits a strong nonlinear phenomenon with a double loop, suggesting that a second, higher-frequency flow structure is embedded in the main flow. The linear model clearly is incapable of predicting the correct flow behavior.

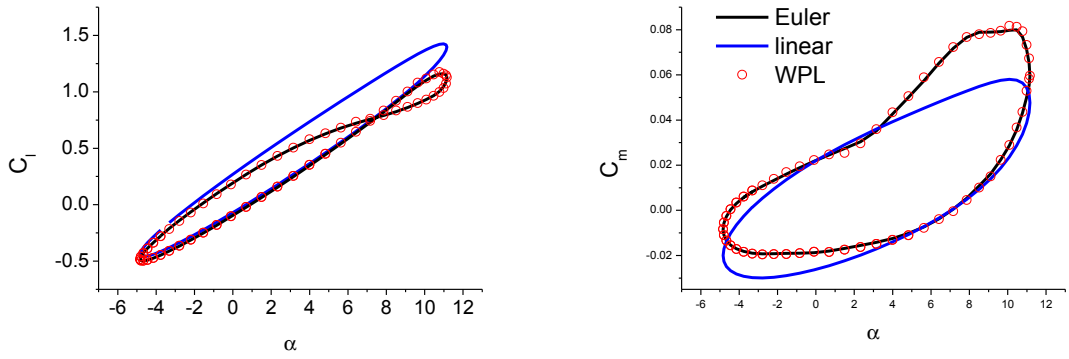


Figure 11. Comparison of Euler, nonlinear WPL-based model and linear solutions of the CT2 airfoil at different pitching amplitudes, with $\alpha_0 = 8°$.

A snapshot of the detailed flowfield is displayed in Figure 12 for the training condition ($\alpha_0 = 8^o$); no discernable differences between the Euler and WPL-based solutions can be found.

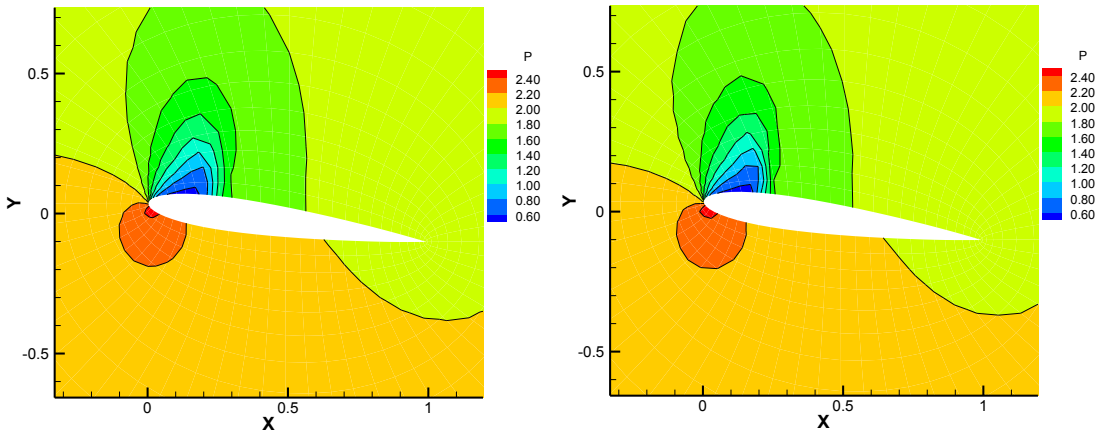


Figure 12. Pressure contours of AGARD CT2 at a time instant, *T*/4, for the airfoil pitching at $\alpha_0 = 8°$ (Left: Euler, Right: WPL).

Again, we are interested in seeing how the model so constructed performs outside the training trajectory for the CT2 problem. With the previously trained model for the trajectory prescribed by Eq. (47) with $\alpha_0 = 8°$, and $\alpha_m = 3.16°$, we now apply it to a new motion defined by Eq. (48), which contains high (triple)-frequency contents resulting from the cubic function and inflection points at t=0, T/2 and T.

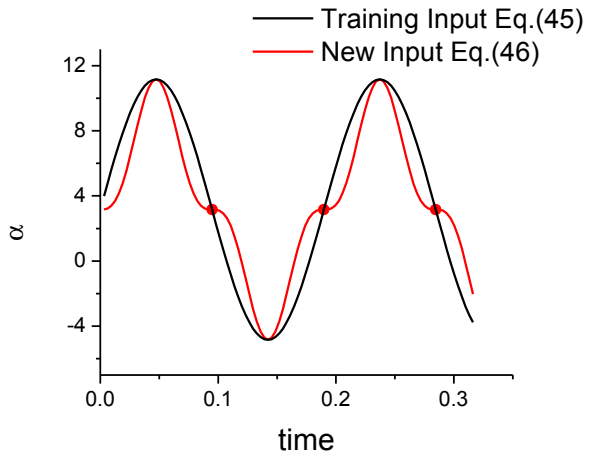


Figure 13. New input trajectory defined in Eq. (46).

A vastly different behavior is revealed in both the lift and moment coefficients in Figure 14, showing sharp changes at the mean angle $\alpha_m = 3.16°$ in both upward and downward phases. At the knot point ($\alpha \approx 6^o$) in C_L, where C_L trajectory crosses, C_M however does not find a particularly unusual behavior except having double values.

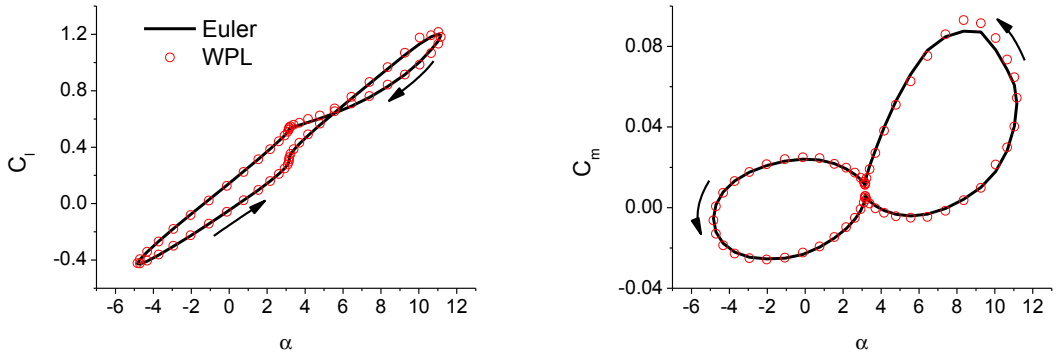
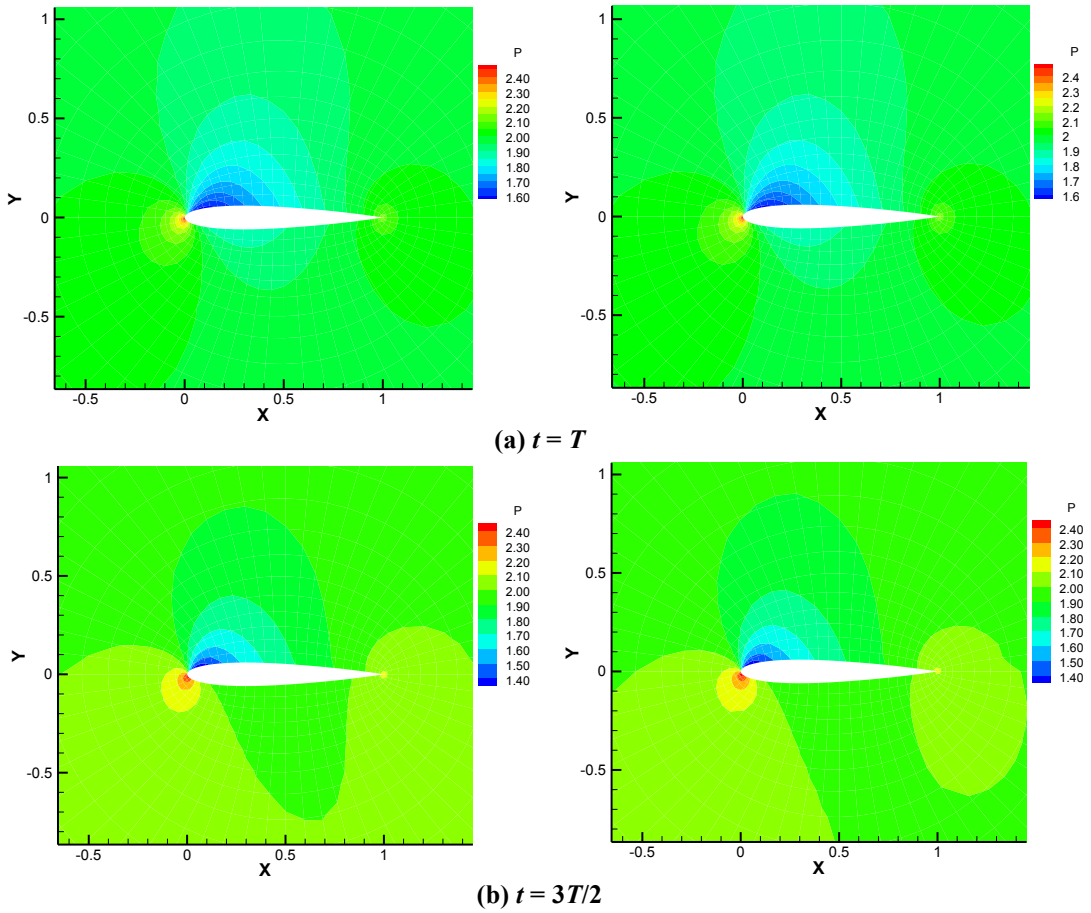


Figure 14. Lift and moment coefficients responding to Eq. (48) input with $\alpha_0 = 8°$.

Finally we show the flowfield at time slices $t = T$ and $3T/2$, both at knot points but with opposite pitching directions, as shown in Figure 15; the results computed by the model are nearly identical to the CFD results in every detail of the flowfield, also indicating a nonsymmetry here even though the airfoil is at the same angle ($\alpha = 3.16°$) at both times.



PFigure 15. Detailed flowfield comparison by the WPL-based model and CFD (Left: Euler, Right: WPL), $\alpha_0 = 10°$.

IV. Concluding Remarks

In many scientific and engineering endeavors, there occurs needs for having a mathematical model that can faithfully describe the physical phenomena under conditions of changing parameters. Examples include design optimization where some geometrical parameters or flow conditions will be required to be changed in order to arrive at an optimal result; similar scenarios also appear when exploring fluid dynamics phenomena resulting from different flow parameters or geometrical effects. A reliable and useful model must be sufficiently accurate to capture important characteristics of the problem under consideration—generally nonlinear in nature and must also remain valid for a sufficiently wide range of parametric changes.

In this paper, we presented an innovative method for constructing a versatile predictive model and we demonstrated for problems for simple nonlinear scalar equation to Euler equations. The method is a nonlinear combination of Taylor series expanded about a set of sampling points (states). The model is trained with a baseline input function and is capable of predicting solutions for different inputs. Built upon our earlier publication where the first-order Taylor expansion was used, we studied the feasibility of including the second-order Taylor terms, which involve complex and computationally expensive Hessian matrices for a system of equations. An approximation of the Hessian matrix was proposed. The accuracy and robustness have been demonstrated for the scalar equation and a two-equation system using this approximate Hessian, and for the Euler equations using the first-order Taylor expansion. Work to implement the approximate Hessin in the Euler system is continuing.

Acknowledgement

Weigang Yao was a NASA Postdoc Fellow supported by the Subsonic Fixed Wing Project, under NASA's Fundamental Aeronautics Program; Mr. William Haller is the Technical Lead of the task. Meng-Sing Liou has been supported by the Subsonic Fixed Wing and Aero Sciences Projects, under NASA's Fundamental Aeronautics Program and Dr. Jeffery Moder is the task Technical Lead for the Aero Sciences Project.

References

1. Rewienski, M. and White, J. "Improving trajectory piecewise-linear approach to nonlinear model order reduction for micro-machine devices using an aggregated projection basis," IEEE Trans. on Computer-Aided Design of Integrated Circuits and System, Vol. 24, No. 2, 2003, 155-170.

2. Gu, C. and Roychowdhury, J., "Model reduction via projection onto nonlinear manifolds with application to analog circuits and biochemical systems," IEEE/ACM International Conference on Computer-Aided Design, 2008, 85-92.

3. Gratton, D. and Willcox, K., "Reduced-order, trajectory piecewise-linear models for nonlinear computational fluid dynamics", 34th AIAA Fluid Dynamics Conference and Exhibit, AIAA 2004-2329.

4. He, J., Saetrom, J., and Durlofsky, L.J., "Enhanced Linearized Reduced-Order Models for Subsurface Flow Simulation," J. Comput. Phys., Vol. 230, No. 23, 2011, 8313-8341.

5. Rugh, W.S. and Shamma, J.S., "Research on gain scheduling," Automatica, Vol. 36, 2000, 1401-1425.

6. Shamma, J.S. and Athans, M., "Gain scheduling: Potential hazards and possible remedies," Control Systems IEEE, Vol.12, No.3, 1992, 101-107.

7. Johansen, T.A. and Murray-Smith, R., "The operating regime approach to nonlinear modeling and control," in Multiple Model Approaches to Modelling and Control (R. Murray-Smith and T.A. Johansen, eds.), Ch. 1, Taylor-Francis, 1997.

8. Nichols, R., Reichert, R. and Rugh, W., "Gain-scheduling for H-Infinity controllers: A flight control example," IEEE Transactions on Control Systems Technology, Vol. 1, No. 2, 1993, 69-79.

9. Marcos, A. and Balas, G.J., "Development of linear-parameter-varying models for aircraft," Journal of Guidance, Control, and Dynamics, Vol. 27, No. 2, 2004, 218-228.

10. Chaturantabut, S., & Sorensen, D. (2010). Nonlinear model reduction via discrete empirical interpolation. SIAM Journal on Scientific Computing, Vol. 32, No. 5, 2737–2764.

11. Carlberg, K., Farhat, C., Cortial, J., and Amsallam, D., "The GNAT method for nonlinear model reduction: Effective implementation and application to computational fluid dynamics and turbulent flows," Journal of Computational Physics, Vol. 242, 2013, 623–647.

12. Yao, W. and Liou, M.-S., "Reduced-order modeling for flutter/LCO using recurrent artificial neural network", 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, AIAA 2012-5446.

13. Buhmann, M.D., Radial Basis Functions: Theory and Implementations, Cambridge University Press, 2003.