

Non-Intrusive Reduced Order Modeling of Convection Dominated Flows Using Artificial Neural Networks with Application to Rayleigh-Taylor Instability

Zhen Gao^{1,*}, Qi Liu¹, Jan S. Hesthaven², Bao-Shan Wang¹,
Wai Sun Don¹ and Xiao Wen³

¹ School of Mathematical Sciences, Ocean University of China, Qingdao, China.

² Mathematics Institute of Computational Science and Engineering, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland.

³ College of Mathematics and Systems Science, Shandong University of Science and Technology, Qingdao, China.

Received 7 April 2020; Accepted (in revised version) 14 December 2020

Abstract. A non-intrusive reduced order model (ROM) that combines a proper orthogonal decomposition (POD) and an artificial neural network (ANN) is primarily studied to investigate the applicability of the proposed ROM in recovering the solutions with shocks and strong gradients accurately and resolving fine-scale structures efficiently for hyperbolic conservation laws. Its accuracy is demonstrated by solving a high-dimensional parametrized ODE and the one-dimensional viscous Burgers' equation with a parameterized diffusion coefficient. The two-dimensional single-mode Rayleigh-Taylor instability (RTI), where the amplitude of the small perturbation and time are considered as free parameters, is also simulated. An adaptive sampling method in time during the linear regime of the RTI is designed to reduce the number of snapshots required for POD and the training of ANN. The extensive numerical results show that the ROM can achieve an acceptable accuracy with improved efficiency in comparison with the standard full order method.

AMS subject classifications: 35L65, 35Q35, 76N15

Key words: Rayleigh-Taylor instability, non-intrusive reduced basis method, proper orthogonal decomposition, artificial neural network, adaptive sampling method.

*Corresponding author. *Email addresses:* zhengao@ouc.edu.cn (Z. Gao), lq1697@stu.ouc.edu.cn (Q. Liu), jan.hesthaven@epfl.ch (J. S. Hesthaven), wbs@stu.ouc.edu.cn (B.-S. Wang), donwaisun@outlook.com (W. S. Don), xiaowen.ouc@163.com (X. Wen)

1 Introduction

In recent years, the Rayleigh-Taylor instability (RTI) [35,40] phenomenon has gained increasing attention in many areas of scientific research, including interstellar medium and galaxy clusters [33], accretion onto the magnetospheres of neutron stars [43,44], and filamentary structure on the Sun [26]. Due to the practical importance, a number of numerical methods have been developed to simulate the RTI, including the flux-corrected transport method [47], the level set method [29], the front tracking method [41], the smoothed particle hydrodynamics method [39], the boundary integral method [14], direct numerical simulations [13], large-eddy simulations [8], and the phase-field method [11,38]. Besides, Zhang et al. [48–50] studied the weakly nonlinear incompressible single-mode RTI and the multi-mode RTI in spherical and planar geometries, and a two-dimensional thin shell model for the nonlinear RTI in spherical geometry is proposed by Zhao et al. [51].

Previous methods are prohibitively expensive when one seeks to reproduce the fine-scale structures of RTI, because they require repeatedly solving a high dimensional system of partial differential equations (PDEs) at a fine grid resolution. To seek alternatives to solve the full problem many times, one can regard the governing equations of RTI as a parameterized time-dependent PDE with parameterized initial conditions. The amplitude of the initial perturbation waves (A) and time (t) are considered as the parameters in two-dimensional parameter space. In this light, the reduced order modeling (ROM) methods [6,27] have been taken into consideration due to its effectiveness in solving the parameterized problems. The main goal of the ROM is to replace the *full-order system* with a *reduced-order model* with a significantly smaller size, which results in a great decrease in the computational cost in CPU times and memory storages, with a controlled loss of accuracy [23].

Featuring an offline-online operational framework, the reduced basis method (RBM) [22,34] is a powerful technique for the ROM methods of the parameterized problems. In general, RBM aims to approximate any member of the solution manifold with a low number of basis functions, the so-called reduced basis (RB), which are extracted from snapshots of the full-order solutions during the offline stage. The full-order system is projected onto the linear subspace spanned by the RB through a projection approach, such as Galerkin projection [17,36]. Although the Galerkin procedure brings high accuracy, it also faces some challenges in computing the projection coefficients for complex nonlinear problems with a non-affine parametric dependence. To avoid this problem, a non-intrusive ROM method [9], in which the original system is only used to generate the snapshots and does not require the projection process [12], is proposed. Necessary techniques have been developed for the non-intrusive ROM methods, among which the proper orthogonal decomposition (POD) [30] is usually applied to extract the RB from the snapshots. In the last decade or so, a variety of methods based on the POD are introduced. In [2,3], Audouze et al. proposed the non-intrusive ROM method based on the POD and radial basis functions (RBFs), where RBFs are used for approximating the solutions in the parameter domain. Guo and Hesthaven [20,21] used the Gaussian process

regression to approximate the coefficients of the ROM for nonlinear structural analysis. Meanwhile, Hesthaven and Ubbiali [23] employed the artificial neural network (ANN) to compute the coefficients of the ROM, but their work focused on the parameterized steady-state partial differential equations. Then Wang et al. [42] employed this approach to solve the one-dimensional parametrized Euler/Navier-Stokes equations to simulate the quasi-1D continuously variable resonance combustor flow. As for the ANN architecture and setup, they used the feed-forward network in the TensorFlow. Readers are referred to [19, 37, 46] for the detailed discussion on the approximation bounds by the ANN for many-variate, real-valued functions f .

In this work, we primarily study a non-intrusive ROM based on the POD and ANN in recovering the solutions with shocks and strong gradients accurately and resolving fine-scale structures efficiently for the two-dimensional hyperbolic conservation laws. To construct the ROM, which is a linear combination of the RB functions, the POD is applied to generate the RB functions from the snapshots. The corresponding coefficients of the RB functions are then computed by training with the ANN, which is the cascade-forward network in the MATLAB. Once the ROM is built, the desired solution with a given parameter can be recovered online efficiently with a slight loss of accuracy. The good performance of the proposed method is demonstrated by solving a high-dimensional parametrized ODE and the one-dimensional Burgers' equation. Then, the simulations of the two-dimensional single-mode compressible RTI are conducted by the proposed method to examine its performance in the early linear and weakly nonlinear (quasi-linear) temporal development of the increasingly complex small scale vortical rollup structures due to the instability of the fluid flow with a large Atwood number. The uniform and adaptive sampling methods are put forward under the consideration of the appearance of the mushroom shaped vertical structures with the increasing time. The number of snapshots required during the early development of the perturbation space (linear regime) of the adaptive sampling method is two times less than the uniform sampling method with equivalent accuracy. Therefore, the adaptive sampling method in time is proposed to decrease the number of samples for POD in the linear regime and the training of the ANN.

The paper is organized as follows. The RBM is presented in Section 2, where the construction of the RB using POD is introduced. In Section 3, the theory and the design of ANN are briefly reviewed. Section 4 illustrates how to construct the ROM based on POD and ANN. Several numerical results are discussed in Section 5 to highlight the accuracy and efficiency of the proposed method. Finally, Section 6 gives conclusions.

2 Reduced basis method (RBM)

The general d -dimensional formulation of the well-posed parameterized time-dependent problem is given by

$$\mathcal{L}[Q(x,t;\nu)] + \mathcal{N}[Q(x,t;\nu)] = S(x,t;\nu), \quad (x,t,\nu) \in \Omega \times \mathcal{T} \times \mathcal{W}, \quad (2.1)$$

Table 1: Three cases of parameter μ and parameter domain \mathcal{P} .

Case	Parameter μ	Parameter domain \mathcal{P}
I	ν	\mathcal{W}
II	t	\mathcal{T}
III	(ν, t)	$\mathcal{W} \times \mathcal{T}$

with some properly defined initial and boundary conditions. Here $\Omega \subset \mathbb{R}^d$ ($d = 1, 2, 3$), $\mathcal{T} \subset [0, T]$ and $\mathcal{W} \subset \mathbb{R}^m$, with m being the number of parameters characterizing the model, represent the space, time and parameters domains, respectively. $\{Q, S\}: \Omega \times \mathcal{T} \times \mathcal{W} \rightarrow \mathbb{R}^n$ denote the parameterized time-dependent solution and the source term, respectively, where n is the dimension of the system (the number of the dependent variables). Moreover, $\mathcal{L}[\bullet; \nu]$ and $\mathcal{N}[\bullet; \nu]$ are the linear and nonlinear operators, respectively, both associated with x and t and characterized by the parameter ν .

After discretizing Eq. (2.1) in Ω by some stable and accurate spatial discretization schemes, one has a system of parameterized ODEs

$$\mathcal{L}_h[Q_h(t; \nu)] + \mathcal{N}_h[Q_h(t; \nu)] = S_h(t; \nu), \quad (t, \nu) \in \mathcal{T} \times \mathcal{W}, \quad (2.2)$$

where $Q_h: \mathcal{T} \times \mathcal{W} \rightarrow \mathbb{R}^M$ is the discrete solution, M is the number of degrees of freedom (DOFs), $\mathcal{L}_h, \mathcal{N}_h$ and $S_h: \mathcal{T} \times \mathcal{W} \rightarrow \mathbb{R}^M$ are the discrete counterparts of the linear operator \mathcal{L} , nonlinear operator \mathcal{N} , and source term S , respectively. In many situations, one has to repeatedly solve the full-order high dimensional nonlinear system of partial differential equations (PDEs) in high resolution with the variation of one or more parameters μ , such as the three cases shown in Table 1. It is clear that one needs to seek an efficient alternative to solve the full-order problem repeatedly, which is a time consuming and laborious task for every different parameter in μ . This is the place where a reduced basis model can play an important role by reducing the size of PDEs in exchange for an acceptable loss of accuracy. Particularly, RBM seeks an approximate solution to Eq. (2.1) as a linear combination of the parameter-independent reduced basis (RB) functions $\{\psi_l\}_{1 \leq l \leq L}$. They are built from a collection of high-fidelity snapshots $\{\bar{Q}_h(\mu_1), \dots, \bar{Q}_h(\mu_N)\}$, where the discrete and finite set of given parameters $\mathcal{P}_N = \{\mu_1, \dots, \mu_N\} \subset \mathcal{P}$ may consist of either a uniform lattice or a randomly generated points over the parameter domain \mathcal{P} [22]. The RB functions $\{\psi_l\}_{1 \leq l \leq L}$ can be generated from several strategies, such as the proper orthogonal decomposition (POD), which will be briefly reviewed in the following Section 2.1. For now, we assume that the RB functions are available and define

$$\mathbb{V}_{rb} = \text{span}\{\psi_1, \dots, \psi_L\} \subset \mathbb{V}_h, \quad (2.3)$$

as the associated RB space of dimension L , where \mathbb{V}_h is a suitable finite element subspace of the Hilbert space \mathbb{V} over Ω . A RB solution $Q_{rb}(x; \mu)$ is sought in the linear subspace

\mathbb{V}_{rb} , that is,

$$Q(x; \mu) \approx Q_{rb}(x; \mu) = \sum_{i=1}^L c_i(\mu) \psi_i(x) \in \mathbb{V}_{rb}, \quad (2.4)$$

where $\vec{c}(\mu) = [c_1(\mu), \dots, c_L(\mu)]^T \in \mathbb{R}^L$ is the vector of the coefficients of the ROM. For simplicity, we denote the column vector gathering the nodal values of ψ_i , $\vec{\Psi}_i = \psi_i(x_j)$, $j = 1, \dots, M$, $i = 1, \dots, L$ and introduce a finite-dimensional RB matrix

$$\Phi = [\vec{\Psi}_1, \dots, \vec{\Psi}_L] \in \mathbb{R}^{M \times L}. \quad (2.5)$$

2.1 Proper orthogonal decomposition (POD)

Let us introduce the notion of a solution manifold [22], i.e., the set of all solutions to the parameterized problem Eq. (2.1) under a continuous variation of the parameter, i.e. $\mathcal{M} = \{Q(\mu) : \mu \in \mathcal{P}\}$ and its discrete form $\mathcal{M}_h = \{Q_h(\mu) : \mu \in \mathcal{P}\} \subset \mathbb{R}^M$. By collecting N snapshots of the discrete solution with N different parameters $\{\vec{Q}_h(\mu_1), \dots, \vec{Q}_h(\mu_N)\}$, corresponding to the finite and discrete parameter set $\mathcal{P}_N = \{\mu_1, \dots, \mu_N\} \subset \mathcal{P}$, one can define the associated subspace

$$\mathcal{M}_\Xi = \text{span}\{\vec{Q}_h(\mu_1), \dots, \vec{Q}_h(\mu_N)\}. \quad (2.6)$$

We shall assume that \mathcal{M}_Ξ provides a good approximation of the discrete solution manifold \mathcal{M}_h , as long as the number of snapshots N is sufficiently large (but is still typically much smaller than the dimension M of the full-order space). Then, we seek to find a parameter-independent RB function for \mathcal{M}_Ξ , i.e., a collection of RB functions $\{\psi_1, \dots, \psi_L\} \subset \mathcal{M}_\Xi$, with $L \ll M$, so that the associated linear space constitutes a low-rank approximation of \mathcal{M}_Ξ . To this end, let us recall the way to construct the RB by POD introduced in [7]. Analogous decompositions such as the principal component analysis and the Karhunen-Loeve expansion of a parametric/random field are alternative ways. The first step is to arrange the nodal values of the N snapshots of the discrete solutions $\{\vec{Q}_h(\mu_1), \dots, \vec{Q}_h(\mu_N)\}$ column-wise into a snapshot matrix

$$\mathbf{Q} = [\vec{Q}_h(\mu_1), \dots, \vec{Q}_h(\mu_N)] \in \mathbb{R}^{M \times N}. \quad (2.7)$$

Then each column vector of \mathbf{Q} can be expanded as a linear combination of the RB functions $\vec{\Psi}_j$ ($j = 1, \dots, L$) in the RB matrix Φ , i.e.,

$$\vec{Q}_h(\mu_i) \approx \Phi \vec{c}(\mu_i), \quad i = 1, \dots, N, \quad (2.8)$$

with $\vec{c}(\mu_i) = [c_1(\mu_i), \dots, c_L(\mu_i)]^T$. The next step is to find the set of orthonormal basis $\{\vec{\Psi}_1, \dots, \vec{\Psi}_L\}$ that in the least square sense minimizes the error,

$$e = \sum_{i=1}^N \left\| \vec{Q}_h(\mu_i) - \sum_{j=1}^L c_j(\mu_i) \vec{\Psi}_j \right\|^2. \quad (2.9)$$

Once the basis functions $\vec{\Psi}_j$ are found for any given μ_i , the coefficients $c_j(\mu_i)$ can be easily found as

$$c_j(\mu_i) = \vec{Q}_h(\mu_i)^T \vec{\Psi}_j = \vec{\Psi}_j^T \vec{Q}_h(\mu_i). \quad (2.10)$$

To generate the RB function $\vec{\Psi}_j$, one can regard the j -th principal component $\vec{\Psi}_j$ as a linear combination of each row vector of \mathbf{Q} with the normalized unknown coefficients vector $\vec{v}_j \in \mathbb{R}^N$, which is the j -th eigenvector of the correlation matrix $\mathbf{D} = \mathbf{Q}^T \mathbf{Q}$ with the $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_L$, i.e.

$$\mathbf{D} \vec{v}_j = \lambda_j \vec{v}_j, \quad (2.11)$$

then the RB function can be expressed as

$$\vec{\Psi}_j = \lambda_j^{-1/2} \mathbf{Q} \vec{v}_j, \quad (2.12)$$

where L , the number of RB functions, is determined according to the following criteria

$$\frac{\sum_{j=1}^L \lambda_j}{\sum_{j=1}^N \lambda_j} \geq C, \quad (2.13)$$

where $0 < C < 1$ is a user-defined threshold ratio, which can be interpreted as the amount of energy of the PDEs system being retained by the RB functions. $C = 0.99999$ (Sections 5.1 and 5.2) and $C = 0.9999$ (Section 5.3) are used in this work.

3 Artificial neural networks

An artificial neural network (ANN) [25] is a network of simple elements called neurons, which receive input, change their internal state (activation) according to the input, and produce output depending on the input and activation. It mimics the working of a bio-physical neuron with inputs and outputs but is not a biological neuron model. The network forms a directed, weighted graph by connecting the output of certain neurons to the input of other neurons. The weights can be modified by a learning process, as described below. In this work, we will focus on a multi-layer perceptron (MLP) feed-forward network.

3.1 MLP neural networks

The MLP neural network [32], which is one of the most widely used neural network models, can learn an input-output mapping without needing to express the mathematical equations that describe the mapping explicitly. For easy discussion of the MLP ANN, a five-layer MLP neural network model including an input layer, three hidden layers, and an output layer is shown in Fig. 1, which also demonstrates several notations as

- $\vec{X} = \{x_j, j = 1, \dots, M\}^T$, $\vec{O} = \{o_k, k = 1, \dots, L\}^T$: The input and output vectors of the input and output layers, respectively;

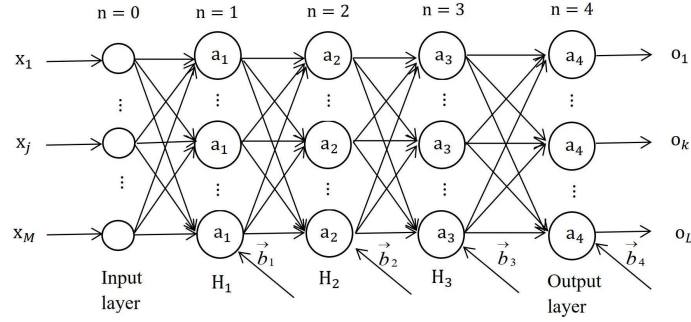


Figure 1: A five-layer MLP neural network model.

- $\vec{b}_n, (n=1,2,3,4)$: The vectors of the bias term of the hidden layers and output layer;
- $a_n, (n=1,2,3,4)$: The activation functions of the hidden layers and output layer;
- \mathbf{W} : The weight matrix between any two layers.

Remark 3.1. The purpose of introducing the activation function is to introduce non-linearity into the model. If there is no activation function, no matter how many layers the neural network has, it will eventually be a linear mapping. The commonly used activation functions are sigmoid function, tanh function, ReLU function, etc. In this work, we use the activation functions that come with the neural network package of MATLAB. For example, the sigmoid function

$$f_{sig}(x) = \frac{1}{1+e^{-x}}$$

is used for the output of the hidden layer. The tanh function

$$f_{tan}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

performs better when the characteristics of the solution are significantly different. Besides, it continuously expands the characteristic effect during the iterations. Usually, the activation function of the output layer (a_4 in Fig. 1) is an identification function. Thus the activation and output of a neural coincides, i.e. $a_4(x) = x$. Based on our extensive numerical experiments, the activation functions $a_1(x) = a_3(x) = f_{tan}(x)$, $a_2(x) = f_{sig}(x)$ generally perform better than other choices.

3.2 Algorithms of MLP neural networks

Given the MLP output vector $\vec{O} = \{o_k, k=1, \dots, L\}^T$ and the expected output vector $\vec{T} = \{t_k, k=1, \dots, L\}^T$ of P training samples, the goal of the MLP neural network is to minimize

the total error

$$E = \frac{1}{L} \sum_{p=1}^P \left\| \vec{T}^p - \vec{O}^p \right\|_2^2 = \frac{1}{L} \sum_{p=1}^P \sum_{k=1}^L (t_k^p - o_k^p)^2, \quad (3.1)$$

by adjusting the elements of the weight matrix \mathbf{W} in a feedback loop until the error reaches an acceptable accuracy $\varepsilon \leq 10^{-5}$.

In order to achieve the objective, two processes are employed in the learning rule of the neural network: the forward calculation process and the error back propagation process.

- *The forward calculation process of signals.*

This process describes how to obtain the output $\vec{O} = \{o_k, k=1, \dots, L\}^T$ through the hidden and output layers from the given input \vec{X} .

In general, the output of the $(n+1)$ -th layer can be expressed as

$$\vec{O}_{n+1} = a_{n+1}(\mathbf{W}_{n,n+1} \vec{O}_n + \vec{b}_{n+1}),$$

where a_{n+1} is the activation function of the $(n+1)$ -th layer, $\mathbf{W}_{n,n+1}$ is the weight matrix between the n -th and $(n+1)$ -th layers, \vec{b}_{n+1} is the vector of the bias term of the $(n+1)$ -th layer. The initial output of the input layer is $\vec{O}_1 = \vec{X}$. Note that the weight matrix \mathbf{W} , and the vector of the bias terms \vec{b} are initialized randomly by the system.

- *The error back propagation process.*

If the error E (3.1) does not reach the specified user-defined tolerance error ϵ , then the weight matrix \mathbf{W} and vector of the bias terms \vec{b} of each layer will be adjusted via the steepest descent method backwardly until reaching the input layer. The changes of the weight w_{ij} and bias term b_i , for example, can be expressed as

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}}, \quad \Delta b_i = -\eta \frac{\partial E}{\partial b_i},$$

where $\eta > 0$ is the learning rate which is employed to control the convergence speed of the training. The weight will then be adjusted as $w_{ij} \leftarrow w_{ij} + \Delta w_{ij}$ for the next forward calculation process. The training is performed for a sufficient number of epochs to obtain a converged network.

These two processes are carried out iteratively until the tolerance error ε or the desired output has been reached. Then the learning process is completed.

3.3 The design of MLP neural networks

In 1989, Robert Hecht-Nielsen [15] proved that a continuous function in any closed interval could be approximated by a MLP neural network with one hidden layer. In theory,

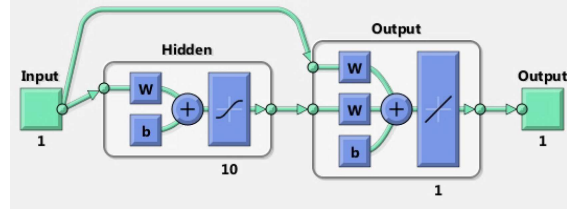


Figure 2: The cascade-forward network with three-layers.

any function from one finite-dimensional space to another can be represented by a feed-forward network with at least one hidden layer, but it can be shown that some functions can only be learned with a single hidden layer and an infinite number of neurons, meaning it is more efficient to learn them with two (or more) hidden layers. Thus, to improve the accuracy of the simulations, a MLP neural network with three hidden layers is used in Sections 5.1 and 5.3 respectively.

Moreover, the number of nodes of the hidden layers could influence the performance of the neural network once the number of layers is determined. Enough neurons of the hidden layer are the premise to ensure that the nonlinear network approximates any curve, but too many neurons of the hidden layer will slow down the convergence rate. According to [16], the number of the nodes of the hidden layer can be estimated by

$$l = \sqrt{n+m} + b, \quad (3.2)$$

where n and m are the numbers of nodes of the input and output layers, respectively, and $b \in [1, 10]$ is a constant.

The feed-forward network and the cascade-forward network are the most commonly used in function approximation. As mentioned in [4], the difference between them is that the cascade-forward network includes a weight connection from the input to each layer and from each layer to the successive layers, which might improve the speed of learning the desired mapping relationship of the network. The cascade-forward model is similar to the feed-forward network, but the main element of this network is that each layer of neurons relates to all previous layer of neurons. In Fig. 2, the cascade-forward network [18, 24] created by the MATLAB command *newcf* is used in this work. We refer to [1] for the detailed discussion on the approximation theory of DNNs and their practical performance of different widths and depths on a variety of test functions.

4 Reduced basis methods using neural networks

As for approximating the ROM coefficients, one can use the alternative methods, e.g., Gaussian process regression [20, 21], radial basis function (RBF) approximations [2], and Galerkin projection method [17, 36]. The advantage of an ANN over Gaussian process regression is the much higher expressibility of the ANN. When the problem is sufficiently

Table 2: The training input and the target output of the three cases.

Case	Training input μ_{tr}	Target output
I	$v_{tr} = \{v_j: j=1, \dots, m\}$	$\Phi^T \vec{Q}_h(v)$
II	$t_{tr} = \{t_i: i=1, \dots, n\}$	$\Phi^T \vec{Q}_h(t)$
III	$D_{tr} = \{(v, t): v \in v_{tr}, t \in t_{tr}\}$	$\Phi^T \vec{Q}_h(t; v)$

simple, a Gaussian process regression may indeed work well and is often more efficient. However, for sufficiently complex problems, e.g., in the time-dependent fluid dynamics, our experience is that only ANN has sufficiently strong resolution power. RBF based techniques are, in many ways, a simpler version of a Gaussian regression approach, and, like the Gaussian regression, require a larger number of nodes to be efficient, having an adverse impact on the computational efficiency while having known conditioning issues that may impact the overall accuracy. The classical Galerkin projection method usually faces some challenges in computing the projection coefficients for complex nonlinear problems with a non-affine parametric dependence, which can be solved efficiently by the ANN method [23]. Furthermore, the ANN method is convenient for users due to the “plug-and-play” aspect.

The computational efficiency of the RBM relies on the decoupling of the offline and online stages of calculation. As discussed earlier, the RB functions are prepared offline from the high-fidelity snapshots of the *full-order solutions* and are independent of the parameters μ . The *reduced-order solution* for a new given parameter value can be recovered quickly in the online stage. To realize this purpose, one aims to seek a function $\vec{d}(\mu)$ that approximates the RB coefficients $\vec{c}(\mu)$, such that,

$$\vec{d}(\mu) \approx \vec{c}(\mu) \in \mathbb{R}^L, \quad \mu \in \mathcal{P}. \quad (4.1)$$

In this work, a MLP neural network in Fig. 2 with the input μ and the target output $\vec{c}(\mu)$ is trained to obtain the mapping function $\vec{d}(\mu)$ for recovering the reduced-order solutions \vec{Q}_{rb} . The resulted network can output an approximated coefficients $\vec{d}(\mu^*)$ *online* for any given parameter $\mu^* \in \mathcal{P}$. Finally, the associated reduced-order solution is computed by

$$\vec{Q}_{rb}(\mu^*) = \Phi \vec{d}(\mu^*). \quad (4.2)$$

For clarity, Table 2 presents the training data set used for the training of the ANN. It is worth mentioning that the discrete training inputs $D_{tr} = \{(v, t)\}$ in Case III are generated by the tensor product between a point-set in the parameter and time domains. Therefore, Case III is computationally expensive. To reduce this cost, an adaptive sampling method is proposed in this study. An alternative way (such as two-step POD method) can be found in [42].

Armed with the approximated reduced basis solution of size L , which is much smaller than the size of the full solution N , that is, $L \ll N$, ANN can be trained with a much

greater efficiency, especially in a large dataset generated by many full solution snapshots in the high dimensional parameter space μ . Since we are interested in solving a solution with discontinuities and fine-scale structures, it is typically required that N be a large number in order to capture shocks essentially non-oscillatory and to resolve the fine-scale structures efficiently.

5 Numerical results

In this section, the accuracy of the proposed method is verified first by a high-dimensional ordinary differential equation (ODE) and the one-dimensional viscous Burgers' equation, respectively. Finally, the gravitation forced single-mode compressible Rayleigh-Taylor instability (RTI) phenomenon, which has more complex convection dominated flow structures such as high gradients or discontinuities and strong vorticity, is considered to show the good performance of the proposed method with the uniform and adaptive sampling methods. In this work, the following relative error is employed,

$$\epsilon = \frac{\|\vec{Q}_h(\mu) - \vec{Q}_{rb}(\mu)\|_\infty}{\|\vec{Q}_h(\mu)\|_\infty}. \quad (5.1)$$

5.1 HIRES problem

To demonstrate the accuracy of the proposed method in the time domain, we consider a high-dimensional system of ODE - HIRES problem, which originates in plant physiology and describes how light is involved in morphogenesis. It is a stiff system of eight nonlinear ODEs, which are given by

$$\frac{d\vec{Q}}{dt} = f(\vec{Q}), \quad 0 \leq t \leq 5, \quad \vec{Q}(0) = (1, 0, 0, 0, 0, 0, 0, 0.0057)^T, \quad (5.2)$$

with $\vec{Q} = (Q_1, \dots, Q_8)^T \in \mathbb{R}^8$, and $f(\vec{Q}) = \mathbf{M}\vec{Q} + \vec{b}$, where $\vec{b} = (\mu_{10}, 0, 0, 0, 0, -z, z, -z)^T \in \mathbb{R}^8$ with $z = \mu_7 Q_6 Q_8$,

$$\mathbf{M} = \begin{pmatrix} -\mu_1 & \mu_2 & \mu_3 & & & & & 0 \\ \mu_1 & -\mu_4 & & & & & & 0 \\ & & -\mu_6 & \mu_2 & \mu_5 & & & 0 \\ & \mu_3 & \mu_1 & -\mu_8 & & & & 0 \\ & & & & -\mu_{11} & \mu_2 & \mu_2 & 0 \\ & & & \mu_9 & \mu_1 & -\mu_2 & \mu_9 & 0 \\ & & & & & & -\mu_{12} & 0 \\ & & & & & & \mu_{12} & 0 \end{pmatrix} \in \mathbb{R}^{8 \times 8}. \quad (5.3)$$

Here $\vec{\mu}_{tr,i} = \langle \mu_i \rangle (1 + 0.1 \xi_i)$, $i = 1, \dots, 12$ are the random parameters. The expectation values are $\langle \vec{\mu} \rangle = (1.71, 0.43, 8.32, 8.75, 0.035, 10.03, 280, 1.12, 0.69, 0.0007, 1.745, 1.81)$ and $\vec{\xi} = (\xi_1, \dots, \xi_{12})$ are uniformly distributed random variables in $[-1, 1]$.

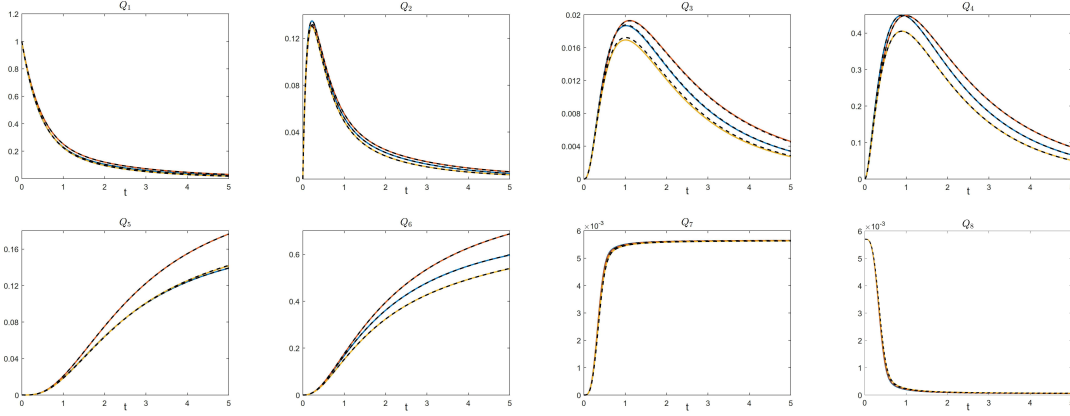


Figure 3: (Color online) Comparison between the full-order solutions (solid lines) and the corresponding reduced-order solutions (dashed lines) with three different parameters of μ .

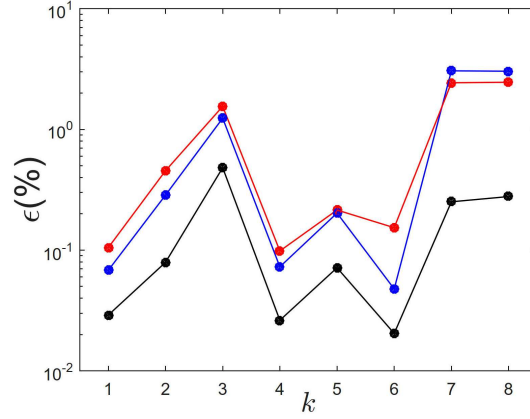


Figure 4: (Color online) The percentage of relative error ϵ of Q_k , ($k=1, \dots, 8$) for three different parameters of μ .

In this study, $L=10$ RB functions are generated by performing the POD on $N_{tr}=1997$ snapshots, which are the numerical solutions of Eq. (5.2) advanced in time by the Runge-Kutta method (MATLAB command `ode15s`) over $t \in [0,5]$. We employ the $N_{tr}=1997$ snapshots as the training data, and a five-layer MLP neural network with 15 neurons in each hidden layer is constructed to obtain the approximated RB coefficients \vec{d} .

The reduced-order solutions with three randomly chosen parameters of μ are recovered and shown in Fig. 3, where the solid and dashed lines represent the full-order and reduced-order solutions, respectively. The reduced-order solution agrees well with the corresponding full-order solutions except for the solution of Q_3 . Furthermore, Fig. 4 shows the corresponding relative errors of the eight solutions Q_k , ($k=1, \dots, 8$), one can easily find that all the relative errors are less than 3%.

5.2 One-dimensional viscous Burgers' equation

To demonstrate the accuracy of the proposed method in the time and parameter domains, the one-dimensional viscous Burgers' equation with parameterized diffusion coefficient

$$Q_t + \left(\frac{Q^2}{2} \right)_x = \frac{\nu}{50\pi} Q_{xx}, \quad (x, t, \nu) \in (-1, 1) \times (0, 1] \times [1, 7.5],$$

$$Q(-1, t; \nu) = Q(1, t; \nu) = 0, \quad Q(x, 0; \nu) = -\sin(\pi x),$$

is considered. In the offline stage, the full-order solutions are calculated by a finite difference method (the first order Euler forward difference in time with $\Delta t = 0.001$ and the second order finite difference scheme for the convection and diffusive terms) with $N_x = 201$ uniformly spaced grid points ($\Delta x = 0.01$), and $N_\nu = 27$ uniformly distributed parameters ν , i.e. $\nu_{tr} = \{1, 1.25, 1.5, \dots, 7.25, 7.5\}$. For the 1000 time steps saved, $N_t = 40$ uniformly distributed samples are included in the training data set, i.e. $t_{tr} = \{0.025, 0.05, 0.075, \dots, 0.975, 1\}$. The corresponding $N_{tr} = N_\nu \times N_t = 1080$ full-order solutions are used as snapshots for constructing the RB functions and as training data for the MLP neural network with two hidden layers with 12 neurons in each layer.

A set of $L = 6$ RB functions are extracted by the POD, as shown in Fig. 5. Since the solution is anti-symmetric (see Fig. 8), the RB functions are also anti-symmetric. Away from the high gradient of the solutions at around $x=0$, the solution is a piece-wise smooth function; the corresponding RB functions are also smooth with slow variations in space.

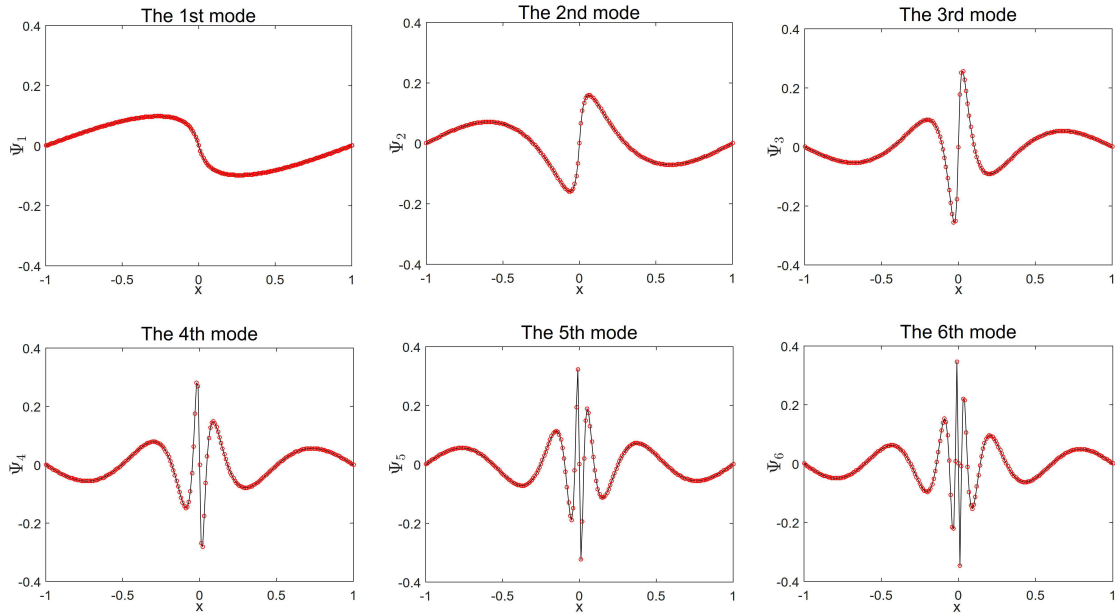


Figure 5: (Color online) The first sixth RB functions for the parameterized viscous Burgers' equation.

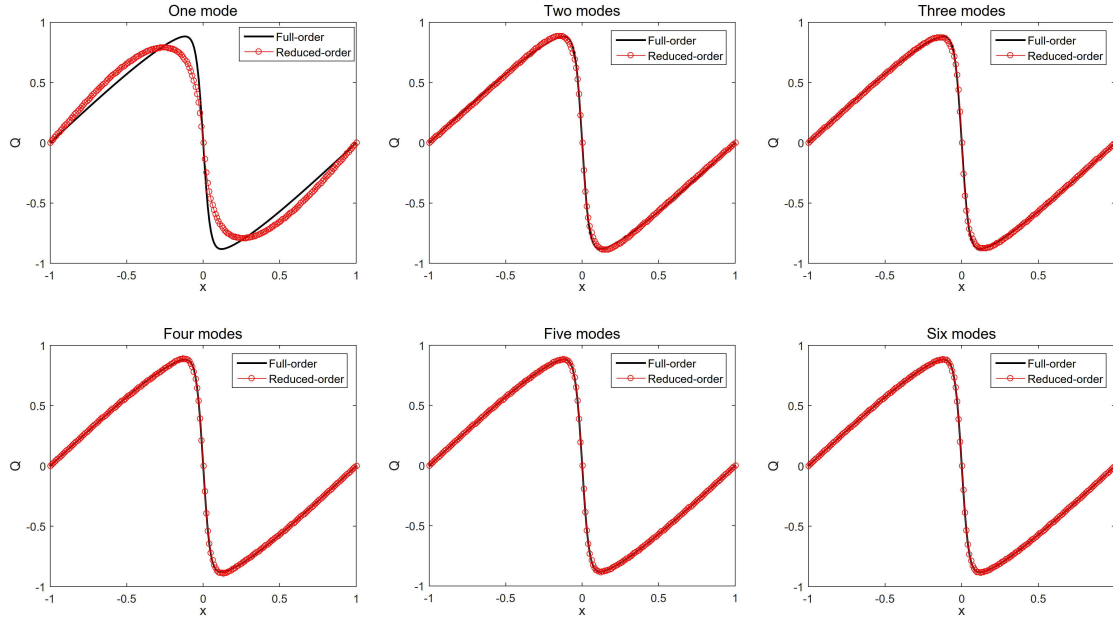


Figure 6: (Color online) The reduced-order solutions with $\nu = 3.125$ at time $t = 0.52$, which are recovered by adding the first mode up to the sixth mode successively and the corresponding coefficients $\vec{c} = (8.0411, -1.4144, -0.1354, 0.1555, -0.0570, 0.0142)^T$.

In the neighborhood of $x = 0$ where the solution exhibits a high gradient, however, the corresponding RB functions are highly oscillatory in order to capture the high gradient. The oscillatory nature of each mode tends to cancel each other out when a sufficient number of modes are summed together in order to obtain an essentially piece-wise smooth function on each side of the high gradient.

To better illustrate the effect of the RB functions in approximating the full-order solutions, the reduced-order solutions with $\nu = 3.125$ at time $t = 0.52$, which are recovered by adding the first mode up to the sixth mode successively, are given in Fig. 6. The coefficients \vec{c} of the first six RB functions decrease sharply with the increasing mode number. It is clear to observe that the recovered solutions with the first two modes already agree quite well with the full-order solutions. To be more specific, the absolute point-wise errors shown in Fig. 7 decrease with the increasing mode number even at the location of the high gradient near $x = 0$.

To further validate the accuracy of the reduced model, the reduced-order solutions (dashed lines) recovered at four different times $t = \{0.27, 0.52, 0.73, 1.00\}$ with four different diffusion coefficients $\nu = \{1.125, 3.125, 5.125, 7.125\}$ reach a good agreement with the corresponding full-order solutions (solid lines) in Fig. 8, which are comparable to those in [20], confirming the accuracy of the proposed method again. The corresponding L_∞ errors in the time domain $[0, 1]$ are shown in Fig. 9, which demonstrates that the maximum point-wise error is less than 10^{-2} .

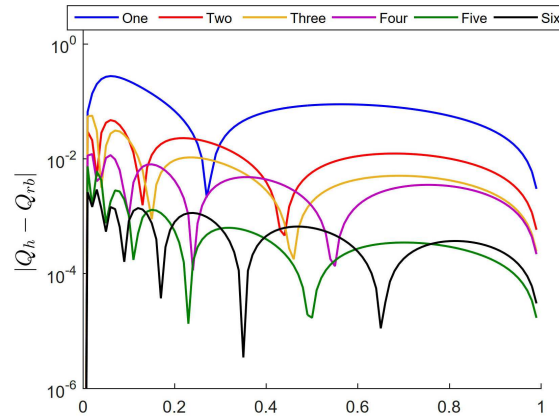


Figure 7: (Color online) The absolute point-wise errors $|Q_h - Q_{rb}|$ between the full-order solutions and the reduced-order solutions with $\nu = 3.125$ at time $t = 0.52$, which are recovered by adding the first mode up to the sixth mode successively.

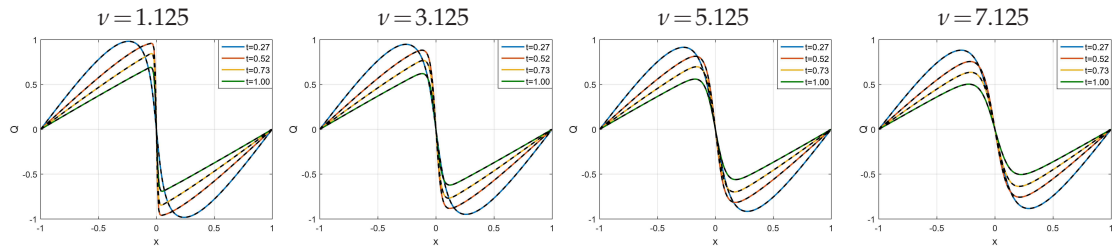


Figure 8: (Color online) Comparison between the full-order solutions (solid) and the corresponding reduced-order solutions (dashed) at times $t = \{0.27, 0.52, 0.73, 1.00\}$ with four different parameters of ν .

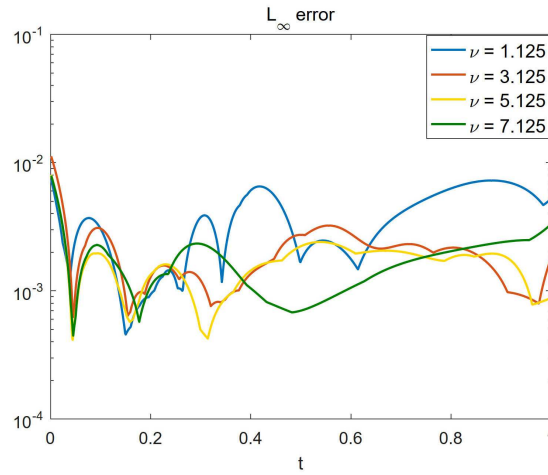


Figure 9: (Color online) The L_∞ error between the full-order solutions and the reduced-order solutions with $\nu = \{1.125, 3.125, 5.125, 7.125\}$ in the time domain $t \in [0, 1]$.

5.3 RTI phenomenon

Finally, we consider the two-dimensional RTI phenomenon with a small single-mode perturbation in the velocity field by the proposed method with uniform and adaptive sampling methods. The governing equations are the two-dimensional hyperbolic conservation laws with source terms, given as

$$\begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix}_t + \begin{pmatrix} \rho u \\ \rho u^2 + P \\ \rho uv \\ (E+p)u \end{pmatrix}_x + \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + P \\ (E+p)v \end{pmatrix}_y = \begin{pmatrix} 0 \\ 0 \\ \rho g \\ \rho gv \end{pmatrix}, \quad (5.4)$$

and the equation of state $P = (\gamma - 1) \left(E - \frac{1}{2} \rho (u^2 + v^2) \right)$ with the specific heat ratio $\gamma = 5/3$ in this work. The ρ, u, v, P and E are the density, velocity in x - and y -directions, pressure, and total energy, respectively. The gravitation constant is normalized as $g = 1$. The initial conditions are

$$(\rho, u, v, P) = \begin{cases} (2, 0, -Ac \cdot \cos(8\pi x), 2y + 1), & y < \frac{1}{2}, \\ (1, 0, -Ac \cdot \cos(8\pi x), y + \frac{3}{2}), & y > \frac{1}{2}, \end{cases}$$

which results in the Atwood number $A_\tau = \frac{1}{3}$, $c = \sqrt{\gamma P / \rho}$ is the sound speed, and A is the amplitude of the perturbation wave which plays an important role in the evolution of RTI phenomenon. The reflective boundary conditions are imposed on the left and right boundaries. At the top boundary, the flow values are set as $\rho = 1$, $P = 2.5$, $u = v = 0$, and at the bottom boundary, $\rho = 2$, $P = 1$, $u = v = 0$. The computational domain is $x \times y = [0, \frac{1}{4}] \times [0, 1]$. The final simulation time is $T = 1.95$.

In this work, the densities of full-order solutions computed by the seventh order WENO-Z scheme [10] with the mesh resolution $N_x \times N_y = 60 \times 240$ are used to train the neural network and verify the accuracy of the proposed method. The key component of the WENO-Z scheme is a polynomial reconstruction procedure that can adaptively switch from a high order polynomial to a nonlinear weighted sum of low order polynomials, where the nonlinear weights are designed based on the local smoothness of the underlying low order polynomials (for details see [5]). Here, three cases are considered:

- **Case (a):** Take the amplitude A as a parameter, i.e., $\mu = A$, and to recover the density ρ at the final time $T = 1.95$ with arbitrary amplitudes $A^* \in [0.025, 0.05]$;
- **Case (b):** Take the time t as a parameter with a fixed amplitude A , i.e., $\mu = t$, and to recover the density ρ at any time $t^* \in [0, 1.95]$;
- **Case (c):** Take both the amplitude A and time t as parameters, i.e., $\mu = (A, t)$, and to recover the density ρ with arbitrary amplitudes $A^* \in [0.025, 0.05]$ at any time $t^* \in [0, 1.95]$.

The evolution of the small smoothly perturbed interface of the RTI phenomenon undergoes three distinct regimes, which are the linear, quasi-linear (weakly nonlinear) and fully nonlinear regimes. At the early time of the instability, the interface grows linearly while maintaining a smooth, simple structure (linear regime). At some point in time, depending on the amplitude of the perturbation wave A and the Atwood number A_τ , the growth of the interface forming spike and bubble and the mixing layer width (distance between the spike and bubble) grows substantially faster than linear (quasi-linear/weakly nonlinear regime). Soon afterward, the exponential growth of the vorticity, due to the baroclinity of a stratified fluid (misalignment of the gradient of pressure and the gradient of density), generates small scales vortical structures along the elongated interface, and the flow becomes turbulence increasingly in nature (fully nonlinear regime). The description of the three regimes can be visualized in Fig. 13.

As a result, the uniformly distributed samples with many snapshots during the early linear regime are not needed to capture the essentially smooth and simple structures of the solution in the early linear regime, while relatively more snapshots are needed to capture the essence of the highly developed complex structures in the quasi- and nonlinear regimes. Therefore, for the time training set t_{tr} in **Case (b)** and **Case (c)**, an adaptive sampling method is designed for the POD and training the ANN with an equivalent accuracy by reducing the number of snapshots needed in the linear regime when the solution is relatively simple and smooth.

Based on the temporal structures of the RTI, (5.5) obtained by the theory of Jacobs and Catton [28,31] is employed to judge whether the nonlinear structures are present or not,

$$L = L_{init} + a \cdot \lambda_l, \quad (5.5)$$

where $a = 0.1$ is used in this study, L_{init} is the initial position of the peak of the spikes, λ_l is the perturbation wavelength. For simplicity, we let S be the position of the peak of the spike and take fewer samples when $S < L$. Otherwise, the same samples as the uniform sampling method are used.

For easy comparison, we refer to the superscript 'U' and 'A' as the variables related to the uniform and adaptive sampling methods. For example, the training data set of the uniform and adaptive sampling methods are denoted as t_{tr}^U and t_{tr}^A , respectively. The reduced-order solutions with the uniform and adaptive sampling methods are referred to as the uniform and adaptive solutions, respectively.

- **Case (a).**

In this case, only the amplitude A of the initial perturbation wave is considered as a parameter. A five-layer neural network is created with 12 neurons in each of the three hidden layers. The number of snapshots used for constructing the RB functions and training the coefficients is $N_{tr} = 57$ with amplitudes

$$\mu_{tr} = A_{tr} = \{(0.0205 : \Delta A : 0.05)\} \text{ except for } \{0.0250, 0.0320, 0.0495\}, \quad \Delta A = 5 \times 10^{-4}.$$

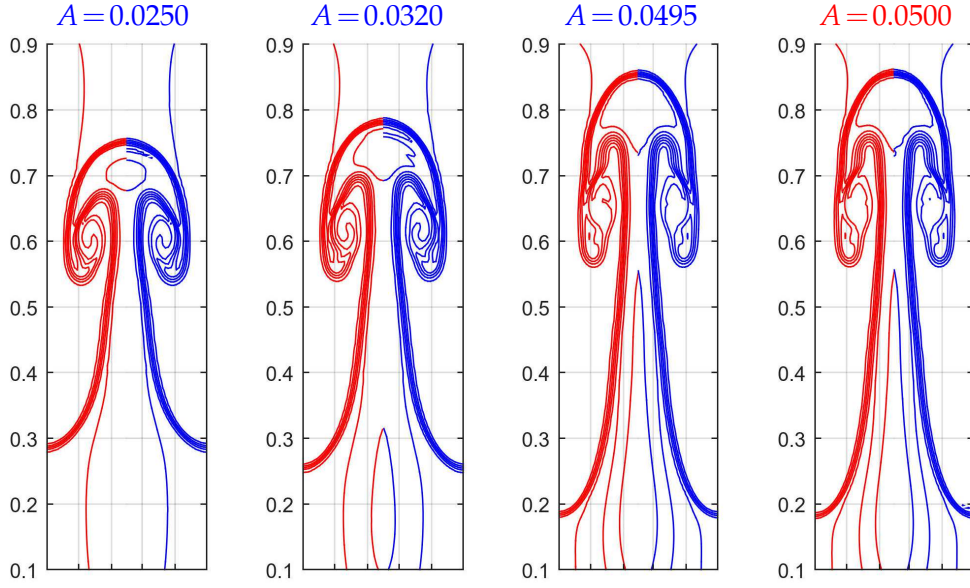


Figure 10: (Color online) The comparison of the density contour lines of the RTI between the full-order solutions (red) and the reduced-order solutions (blue) with amplitudes $A_{test} = \{0.0250, 0.0320, 0.0495, 0.0500\}$.

Remark: $(x_1 : \Delta x : x_2)$ means taking points with a uniform grid spacing Δx in the interval $[x_1, x_2]$.

Here, $L = 14$ RB are selected to obtain the reduced-order solutions. For clarity, the red (blue) titles in the figures below mean data is inside (outside) of the training data set. The reduced-order solutions are recovered with amplitudes $\mu_{test} = A_{test} = \{0.0250, 0.0320, 0.0495, 0.05\}$, and shown in Fig. 10 to verify the accuracy of the ROM. It is clear to find that the ROM can capture the large scale structures of the RTI and maintain the symmetry regardless of the amplitude A is in the training data set or not. But there are subtle differences in some small scale structures due to the truncation of the RB functions.

- **Case (b).**

In this case, we fix the amplitude of the initial perturbation wave as $A_{test} = 0.025$ and regard time t as a parameter. The $N = 66$ full-order solutions at times $t = \{0 : \Delta t : 1.95\}$ with $\Delta t = 0.03$ are computed in the offline stage.

For the two sampling methods, the training data sets are

$$t_{tr}^U = \{0 : 2\Delta t : 1.92\}, \quad t_{tr}^A = \{(0 : 4\Delta t : S_t), (S_t + 2\Delta t : 2\Delta t : 1.92)\},$$

where S_t is the maximum time when $S < L$. As a result, the numbers of snapshots used for the POD and the training of the neural network are $N^U = 33$, $N^A = 29$, respectively. The numbers of RB function are $L^U = 23$, $L^A = 15$, respectively. A five-layer MLP neural network with 13 neurons in each hidden layer is used.

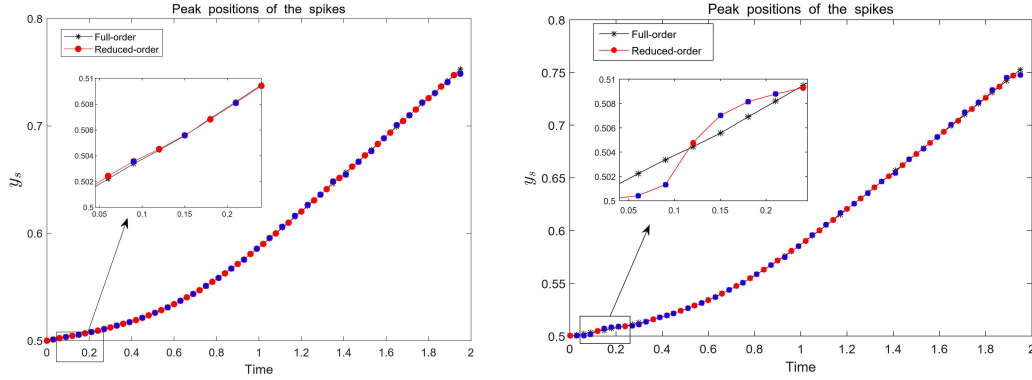


Figure 11: (Color online) (Left) Uniform sampling and (Right) adaptive sampling: The locations of the peak of the spikes at different times $y_s(t)$. '■' represents the time that is outside of the training data set.

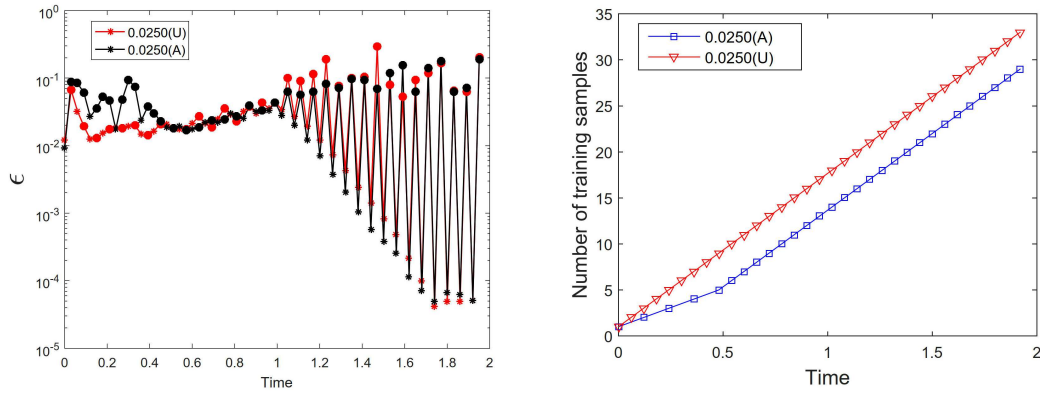


Figure 12: (Color online) (Left) The relative errors between the full-order and reduced-order solutions (black: adaptive solution; red: uniform solution) in time $t \in [0, 2]$: solid circles represent the times that are outside of the training data set. (Right) The comparison of the number of training samples between the uniform sampling (red line) and adaptive sampling (blue line) of the amplitude $A = 0.025$ in time $t \in [0, 2]$.

The time dependent positions of the peak of the spikes $y_s(t)$ play a very important role in analyzing the RTI phenomenon. Therefore, we recover the uniform and adaptive solutions at times $t = \{0 : \Delta t : 1.95\}$ and compute the corresponding peak of the spike $y_s(t)$, as drawn in Fig. 11. Clearly, the time-dependent peak of the spike $y_s(t)$ recovered from the uniform solutions are comparable to the one obtained by the full-order solutions. As for the adaptive sampling method, the difference between the full-order and adaptive solutions are slightly larger ($\leq 10\%$) than those obtained by the uniform sampling method when $t \leq S_t \approx 0.5$. The relative errors between the full-order and reduced-order solutions plotted in the left of Fig. 12 confirm this observation. Furthermore, the adaptive results show a very good agreement with the uniform results when $t \geq S_t$. The number of uniform and

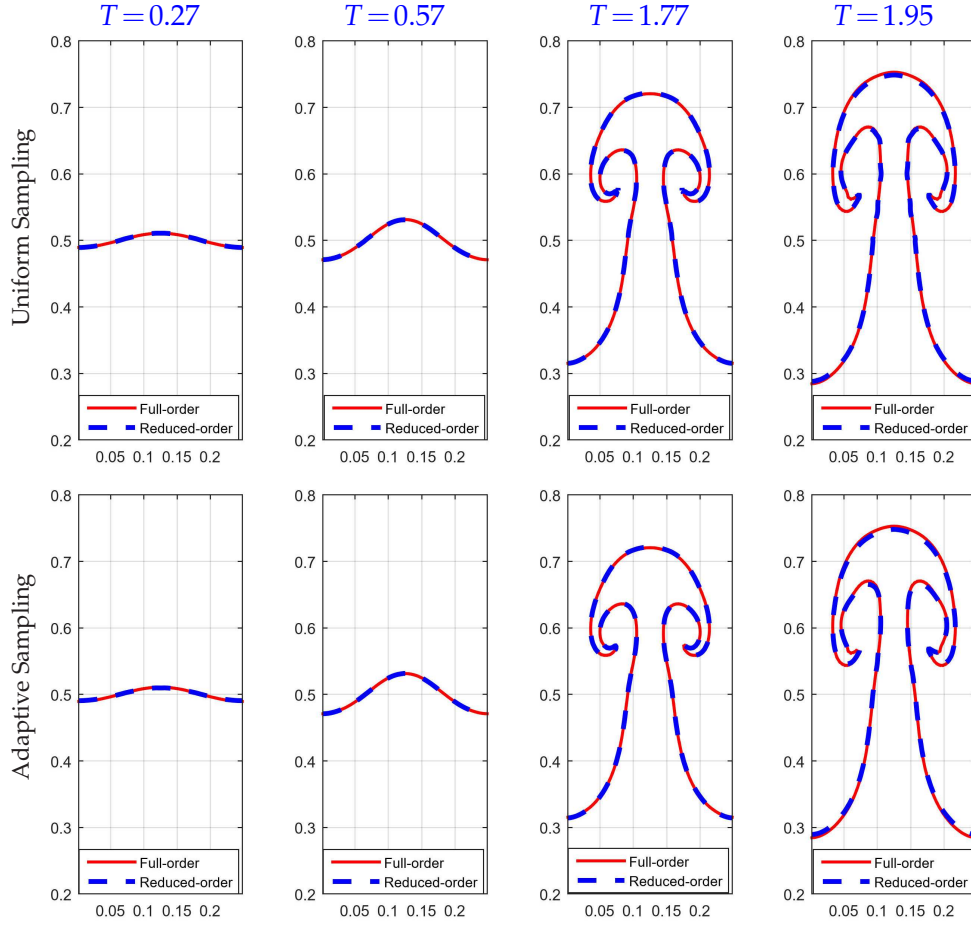


Figure 13: (Color online) The recovered reduced-order density contour lines of $\rho = 1.5$ of the RTI and the corresponding full-order solutions with $A_{test} = 0.025$ at times $t = \{0.27, 0.57, 1.77, 1.95\}$.

adaptive samples distributed in the temporal domain is illustrated in the right of Fig. 12. It is clear that the number of adaptive samples is only half of that needed in the uniform samples when $t \leq 0.5$.

To further observe the large scale structures in detail, four recovered density contour lines of $\rho = 1.5$ at times $t_{test} = t = \{0.27, 0.57, 1.77, 1.95\}$ using uniform and adaptive sampling methods are plotted in Fig. 13. One can observe that the large scale structures of the adaptive solutions agree well with those of the full-order solutions as well as the uniform solutions, even the predicted density $\rho(x, y, t = 1.95)$ estimated at time $t = 1.95$, which is outside the upper bound of the training data set.

- **Case (c).**

With one more parameter considered, more snapshots must be added to the training data set to obtain more RB functions and the corresponding coefficients. Thus,

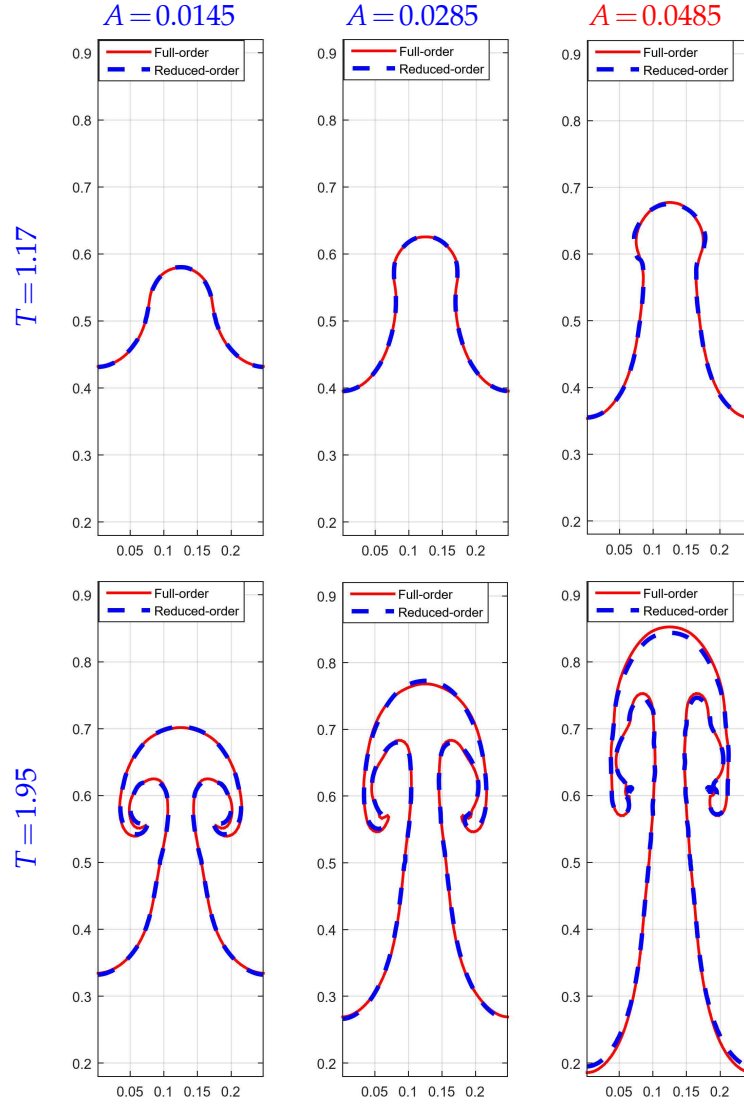


Figure 14: (Color online) Uniform sampling: The recovered reduced-order density contour lines of $\rho=1.5$ of the RTI and the corresponding full-order solutions with parameters in $\{(A, t)_{test}\}$.

$N^u = 5518$ and $N^A = 5008$ samples are prepared in the parameter domain in the offline stage, which is the tensor product of $N_\mu = 89$ amplitudes A and times t . The samples $D_{tr} = \{(\mu_{tr}, t_{tr})\}$ with

$$\begin{aligned} \mu_{tr} &= A_{tr} = \{(0.005 : \Delta A : 0.05)\} \text{ except for } \{0.0145, 0.0285\}, \\ t_{tr}^u &= \{(0 : \Delta t : 1.95)\} \text{ except for } \{0.57, 1.17, 1.77, 1.95\}, \\ t_{tr}^A &= \{(0 : 2\Delta t : S_t), (S_t + \Delta t : \Delta t : 1.92)\}, \end{aligned}$$

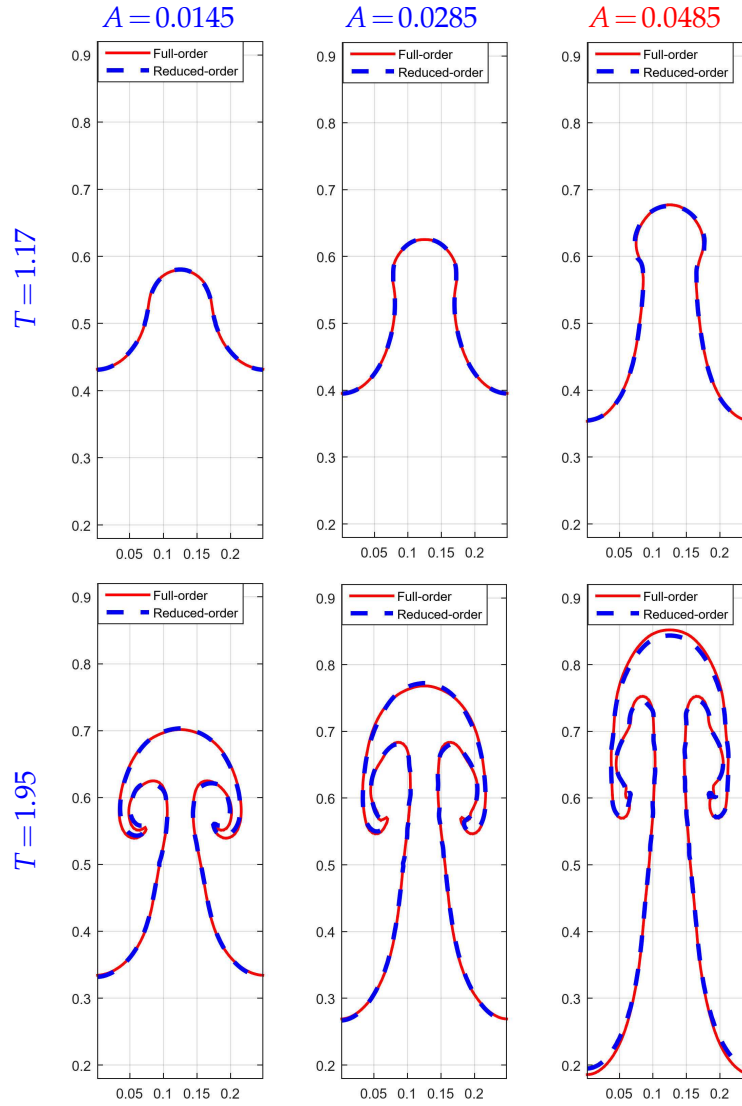


Figure 15: (Color online) Adaptive sampling: The recovered reduced-order density contour lines of $\rho = 1.5$ of the RTI and the corresponding full-order solutions with parameters in $\{(A, t)_{test}\}$.

are used for the POD and training the MLP neural network. The adaptive training data set is half of the size for $t < S_t$ than those for $t > S_t$. $L^u = 32$ and $L^A = 33$ RB functions are generated through the POD respectively. A five-layer network with 13 neurons in each hidden layer is created to approximate the coefficients. Six recovered solutions with the parameters

$$\{(A, t)_{test}\} = \{(0.0145, 1.17), (0.0285, 1.17), (0.0485, 1.17), \\ (0.0145, 1.95), (0.0285, 1.95), (0.0485, 1.95)\},$$

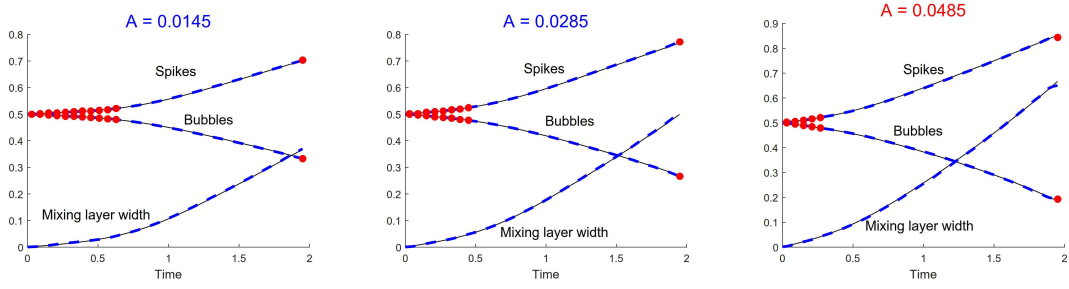


Figure 16: (Color online) Adaptive sampling: The comparison of the positions of the peak of the spikes, the bottom of the bubbles and the mixing layer width between the full-order solutions (solid) and the adaptive solutions (dashed) of the three amplitudes $A_{test} = \{0.0145, 0.0285, 0.0485\}$. '•' represents time that is outside of the training data set.

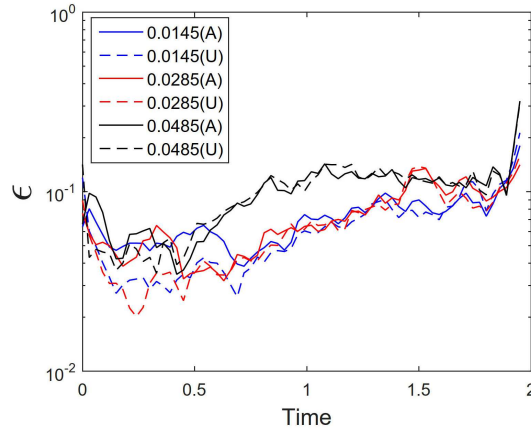


Figure 17: (Color online) The comparison of the relative errors between the uniform sampling (dashed lines) and the adaptive sampling (solid lines) of the three amplitudes $A_{test} = \{0.0145, 0.0285, 0.0485\}$.

are shown in Fig. 14 (uniform sampling method) and Fig. 15 (adaptive sampling method) respectively. One finds that the adaptive solutions are comparable to the uniform solutions and the full-order solutions except for a minor difference at time $t = 1.95$.

In addition, to investigate the accuracy of the adaptive sampling method for other parameters, the temporal history of the peak of the spike $y_s(t)$, the bubble $y_b(t)$, and the mixing layer width $h(t)$ with the three perturbation amplitudes $A = \{0.0145, 0.0285, 0.0485\}$ are drawn in Fig. 16. The adaptive solutions capture the peak of the spikes and bubbles accurately. The corresponding relative errors are also plotted in Fig. 17, which demonstrates that the adaptive solutions are rather similar to the uniform solutions. Furthermore, the number of adaptive samples N^A is proportional to the increasing amplitudes A , as shown in the left of Fig. 18. However, N^A (adaptive sampling) is always still less than N^U (uni-

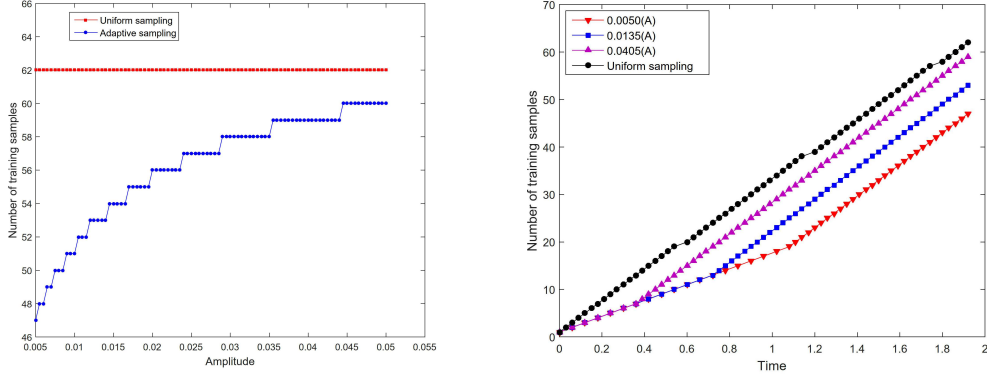


Figure 18: (Color online) The number of training samples (Left) with respect to different amplitudes and (Right) distributed in the temporal domain with respect to the three amplitudes $A_{test} = \{0.005, 0.0135, 0.0405\}$.

form sampling), especially when $A \leq 0.02$. For a clearer comparison, the number of samples distributed in the temporal domain corresponding to the three amplitudes $A = \{0.005, 0.0135, 0.0405\}$ is shown in the right of Fig. 18. It allows that the number of adaptive samples is only half of that needed in uniform samples when $t \leq S_t$. The reason for these results is that the larger the initial perturbation energy (amplitude) is given, the more complex structures are generated.

6 Conclusion

In this study, we develop a non-intrusive reduced basis method for the Rayleigh-Taylor instability by considering the governing equations as the time-dependent parameterized partial differential equation with the parameterized initial conditions. The proposed method extracts a small number of reduced basis functions from the snapshots via the proper orthogonal decomposition to construct the reduced-order model. The undetermined coefficients, which correspond to the reduced basis functions, are approximated by a multi-layer perceptron neural network with two or three hidden layers. The snapshots used for the POD and the training of the neural network can be prepared in advance during the offline stage. The proposed method can avoid the heavy computation of the traditional methods, such that the approximated solutions for any reasonable parameters can be easily recovered online.

The proposed method has been successfully tested on a high-dimensional ordinary differential equation and the one-dimensional viscous Burgers' equation. Furthermore, the RTI phenomenon, where the amplitude and time are considered as the parameters, is simulated by the proposed method with the uniform and adaptive sampling methods. The adaptive method is designed to decrease the number of samples for the POD and training with a minor loss of accuracy after considering the temporal development of

the increasingly complex small scale vortical rollup structures. The results show that the proposed method is able to reduce the model and the CPU time when recovering the desired solution at a reasonable accuracy. In practice, we also find that the number of reduced basis functions, the number of training data, and neurons in the hidden layers of the neural network can affect the accuracy of the proposed method. However, it is out of the scope of this work and will be studied in the future.

Acknowledgments

The authors would like to acknowledge the funding support of this research by the National Natural Science Foundation of China (11871443) and Shandong Provincial Qingchuang Science and Technology Project (2019KJII002). The author (Don) also likes to thank the Ocean University of China for providing the startup funding (201712011) that is used in supporting this work.

References

- [1] B. Adcock, N. Dexter, *The gap between theory and practice in function approximation with deep neural networks*, arXiv:2001.07523v1.
- [2] C. Audouze, F.D. Vuyst and P.B. Nair, *Nonintrusive reduced-order modeling of parametrized time-dependent partial differential equations*, Numer. Meth. Part. D. E. **29**(5) (2013) 1587–1628.
- [3] C. Audouze, F.D. Vuyst and P.B. Nair, *Reduced-order modeling of parameterized PDEs using time-space-parameter principal component analysis*, Internat. J. Numer. Methods Eng. **89**(8) (2009) 1025–1057.
- [4] D. S. Badde, A.K. Gupta and V. K. Patki, *Cascade and Feed Forward Back propagation Artificial Neural Network Models for Prediction of Compressive Strength of Ready Mix Concrete*, IOSR-JMCE. (2015) 1177–1180.
- [5] R. Borges, M. Carmona, B. Costa and W.S. Don, *An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws*, J. Comput. Phys. **227**(6) (2008) 3191–3211.
- [6] T. Bui-Thanh, K. Willcox and O. Ghattas, *Model reduction for large-scale systems with high-dimensional parametric input space*, SIAM J. Sci. Comput. **30**(6) (2008) 3270–3288.
- [7] V. Buljak, *Proper Orthogonal Decomposition and Radial Basis Functions for Fast Simulations. Inverse Analyses with Model Reduction*, Springer Berlin Heidelberg, (2012) 85–139.
- [8] W. Cabot, *Comparison of two- and three-dimensional simulations of miscible Rayleigh-Taylor instability*, Phys. Fluids. **18**(4) (2006): 045101.
- [9] F. Casenave, A. Ern and T. Lelièvre, *A non-intrusive reduced basis method applied to aeroacoustic simulations*, Adv. Comput. Math. **41**(5) (2015) 961–986.
- [10] M. Castro, B. Costa and W.S. Don, *High order weighted essentially non-oscillatory WENO-Z schemes for hyperbolic conservation laws*, J. Comput. Phys. **230**(5) (2011) 1766–1792.
- [11] A. Celani, A. Mazzino, P. Muratore-Ginanneschi and L. Vozella, *Phase-field model for the Rayleigh-Taylor instability of immiscible fluids*, J. Fluid Mech. **622** (2009) 115–134.
- [12] W. Chen, J.S. Hesthaven, J.Q. Bai, Z. Yang and T.H. Yang, *A greedy non-intrusive reduced order model for fluid dynamics*, J. Northwest. Polytech. Univ. (2017).
- [13] A.W. Cook and P.E. Dimotakis, *Transition stages of Rayleigh-Taylor instability between miscible fluids*, J. Fluid Mech. **443** (2001) 69–99.

- [14] L. Duchemin, C. Josserand and P. Clavin, *Asymptotic behavior of the Rayleigh-Taylor instability*, Phys. Rev. Lett. **94**(22) (2005): 224501.
- [15] L.H. Feng and W.H. Hong, *On hydrologic calculation using artificial neural networks*, Appl. Math. Lett. **21**(5) (2008) 453–458.
- [16] H.X. Fu and H. Zhao, *Application design of neural network bsased on MATLAB*, China Machine Press, (2010) 90-91.
- [17] S. Georgaka, G. Stabile, G. Rozza, M.J. Bluck, *Parametric POD-Galerkin Model Order Reduction for Unsteady-State Heat Transfer Problems*, Commun. Comput. Phys. **27**, (2020) 1–32.
- [18] S. Goyal and G.K. Goyal, *Cascade and Feedforward Backpropagation Artificial Neural Network Models For Prediction of Sensory Quality of Instant Coffee Flavoured Sterilized Drink*, Canadian J. Artif. Intell., Mach. Learn. Patt. Recogn. **2**(6) (2011) 78–82.
- [19] I. Gühring, G. Kutyniok and P. Petersen, *Error bounds for approximations with deep ReLU neural networks in $W^{s,p}$ norms*, Anal. Appl. **18**(5) (2020) 803–859.
- [20] M.W. Guo and J.S. Hesthaven, *Data-driven reduced order modeling for time-dependent problems*, Comput. Methods Appl. Mech. Eng. **345** (2019) 75–99.
- [21] M.W. Guo and J.S. Hesthaven, *Reduced order modeling for nonlinear structural analysis using Gaussian process regression*, Comput. Methods Appl. Mech. Eng. **341** (2018) 807–826.
- [22] J.S. Hesthaven, G. Rozza and B. Stamm, *Certified reduced basis methods for parametrized partial differential equations*, Springer, (2016).
- [23] J.S. Hesthaven and S. Ubbiali, *Non-intrusive reduced order modeling of nonlinear problems using neural networks*, J. Comput. Phys. **363** (2018) 55–78.
- [24] <https://www.mathworks.com/help/deeplearning/ref/cascadeforwardnet.html>.
- [25] <https://uijwalkarn.me/2016/08/09/quick-intro-neural-networks/>.
- [26] H. Isobe, T. Miyagoshi, K. Shibata and T. Yokoyama, *Filamentary structure on the Sun from the magnetic Rayleigh-Taylor instability*, Nature, **434** (2005) 478–481.
- [27] K. Ito and S.S. Ravindran, *A reduced-order method for simulation and control of fluid flows*, J. Comput. Phys. **143**(2) (1998) 403–425.
- [28] J.W. Jacobs and I. Catton, *Three-dimensional Rayleigh-Taylor instability Part 1. Weakly nonlinear theory*, J. Fluid Mech. **187** (1988) 329–352.
- [29] X.L. Li, B.X. Jin and J. Glimm, *Numerical study for the three dimensional Rayleigh-Taylor instability through the TVD/AC scheme and parallel computation*, J. Comput. Phys. **126** (1996) 343–355.
- [30] Y.C. Liang, H.P. Lee, S.P. Lim, W.Z. Lin, K.H. Lee and C.G. Wu, *Proper orthogonal decomposition and its applications, part I: theory*, J. Sound Vib. **252**(3) (2002) 527–544.
- [31] J.D. Lindl, P. Amendt and R.L. Berger, *The physics basis for ignition using indirectdrive targets on the National Ignition Facility*, Phys. Plasmas, **11** (2004) 339–491.
- [32] B. Liu and H.X. Guo, *Neural Network Super Learning Manual Based on MATLAB*, Posts and Telecom Press, (2014) 159–168.
- [33] J. Nittmann, S.A.E.G. Falle and P.H. Gaskell, *The dynamical destruction of shocked gas clouds*, Mon. Not. R. Astron. Soci. **201** (1982) 833–847.
- [34] A. Quarteroni, A. Manzoni and F. Negri, *Reduced basis methods for partial differential equations: an introduction*, Springer, **92** (2015).
- [35] L. Rayleigh, *Investigation of the character of the equilibrium of an incompressible heavy fluid of variable density*, Proc. London Math. Soc. **s1-14**(1) (1882) 170–177.
- [36] M.L. Rapún and J.M. Vega, *Reduced order models based on local pod plus Galerkin projection*, J. Comput. Phys. **229**(8) (2010) 3046–3063.
- [37] C. Schwab and J. Zech, *Deep learning in high dimension: Neural network expression rates for generalized polynomial chaos expansions in UQ*, Anal. Appl. **17**(1): (2019) 19–55.

- [38] N. Talat, B. Mavrič, V. Hatić, S. Bajt and B. Šarler, *Phase field simulation of Rayleigh-Taylor instability with a meshless method*, Eng. Anal. Bound. Elem. **87** (2018) 78–89.
- [39] A.M. Tartakovsky and P. Meakin, *A smoothed particle hydrodynamics model for miscible flow in three-dimensional fractures and the two-dimensional Rayleigh-Taylor instability*, J. Comput. Phys. **207** (2005) 610–624.
- [40] G. Taylor, *The instability of liquid surfaces when accelerated in a direction perpendicular to their planes. I*, P. Roy. Soc. A. **201**(1065) (1950) 192–196.
- [41] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas and Y.J. Jan, *A Front-Tracking Method for the Computations of Multiphase Flow*, J. Comput. Phys. **169** (2001) 708–759.
- [42] Q. Wang, J.S. Hesthaven and D. Ray, *Non-intrusive reduced order modeling of unsteady flows using artificial neural networks with application to a combustion problem*, J. Comput. Phys. **384** (2019) 289–307.
- [43] Y.M. Wang and M. Nepveu, *A numerical study of the nonlinear Rayleigh-Taylor instability, with application to accreting X-ray sources*, Astron. Astrophys. **118** (1983) 267–274.
- [44] Y.M. Wang, M. Nepveu and J.A. Robertson, *Further numerical studies of the Rayleigh-Taylor instability in the context of accreting X-ray sources*, Astron. Astrophys. **135** (1984) 66–76.
- [45] Y.J. Xu, F.M. Liang, G. Zhang and H.F. Xu, *Image Intelligent Detection Based on the Gabor Wavelet and the Neural Network*, Symmetry, **8**(11) (2016) 130.
- [46] D. Yarotsky, *Error bounds for approximations with deep ReLU networks*, Neur. Net. **94** (2017) 103–114.
- [47] W.H. Ye, W.Y. Zhang, G.N. Chen, C.Q. Jin and J. Zhang, *Numerical simulations of the FCT method on Rayleigh-Taylor and Richtmyer-Meshkov instabilities*, Chin. J. Comput. Phys. **15**(3) (1998) 278–282.
- [48] J. Zhang, L.F. Wang, W.H. Ye, H.Y. Guo, J.F. Wu, Y.K. Ding, W.Y. Zhang and X.T. He, *Weakly nonlinear incompressible Rayleigh-Taylor instability in spherical and planar geometries*, Phys. Plasmas. **25**(2) (2018): 022701.
- [49] J. Zhang, L.F. Wang, W.H. Ye, J.F. Wu, H.Y. Guo, Y.K. Ding, W.Y. Zhang and X.T. He, *Weakly nonlinear multi-mode Rayleigh-Taylor instability in two-dimensional spherical geometry*, Phys. Plasmas. **25**(8) (2018): 082713.
- [50] J. Zhang, L.F. Wang, W.H. Ye, J.F. Wu, H.Y. Guo, W.Y. Zhang and X.T. He, *Weakly nonlinear incompressible Rayleigh-Taylor instability in spherical geometry*, Phys. Plasmas. **24**(6) (2017): 062703.
- [51] K.G. Zhao, C. Xue, L.F. Wang, W.H. Ye, J.F. Wu, Y.K. Ding, W.Y. Zhang and X.T. He, *Two-dimensional thin shell model for the nonlinear Rayleigh-Taylor instability in spherical geometry*, Phys. Plasmas. **26**(2) (2019): 022710.