

## Article

# Data Reconstruction-Based Two-Step Non-Intrusive Reduced-Order Modeling Using Fourier Transform and Interpolations

Jonggeon Lee <sup>1</sup>, Euiyoung Kim <sup>2</sup> and Jaehun Lee <sup>3,\*</sup><sup>1</sup> Department of Mechanical Engineering, Seoul National University, Seoul 08826, Korea<sup>2</sup> Mechanical Systems Safety Research Division, Department of System Dynamics, Korea Institute of Machinery & Materials, Daejeon 34103, Korea<sup>3</sup> Department of Mechanical, Robotics and Energy Engineering, Dongguk University, Seoul 04620, Korea

\* Correspondence: jaehun@dgu.edu; Tel.: +82-2-2260-3826

**Abstract:** This study presents a data reconstruction-based two-step non-intrusive reduced-order modeling (ROM) based on discrete Fourier transformation (DFT) and proper orthogonal decomposition-radial basis function (POD-RBF) interpolation. To efficiently approximate a system for various parametric inputs, two offline and one online stage are proposed. The first offline stage adjusts and reconstructs sampled data using a scaling factor. During the adjusting procedure, the fast Fourier transform operation is used to transform a domain between the time and frequency, and the POD-RBF interpolation method efficiently generates adjusted data. The second offline stage constructs multiple ROMs in the frequency domain for interpolation with respect to the parameter. Finally, in the online stage, the solution field depending on the changes in input parameters, is approximated using the POD-RBF interpolation and the inverse Fourier transformation. The accuracy and efficiency of the proposed method are verified using the 2-D unsteady incompressible Newtonian fluid problems and are compared to the OpenFOAM software program showing remarkable efficiencies in computing approximated solutions.

**Keywords:** reduced-order model; proper orthogonal decomposition; radial basis function; discrete Fourier transformations; non-intrusive method

**MSC:** 70-08

**Citation:** Lee, J.; Kim, E.; Lee, J. Data Reconstruction-Based Two-Step Non-Intrusive Reduced-Order Modeling Using Fourier Transform and Interpolations. *Mathematics* **2022**, *10*, 3738. <https://doi.org/10.3390/math10203738>

Academic Editor: Miaojuan Peng

Received: 15 September 2022

Accepted: 9 October 2022

Published: 11 October 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Although computational science has been actively studied, a high computational cost is still required to solve complex phenomena or large-scale system problems. Furthermore, research fields that involve iterative calculations require expensive computational resources and time. Therefore, studies on reduced order modeling (ROM) have gained much attention in various fields. Numerous ROM methods have been explored depending on the requirements of research fields, such as vibration/acoustic systems [1], molecular systems [2], computational fluid dynamics [3–6], microelectromechanical systems [7], weather prediction [8], structure systems [9–13], etc.

Among the various model reduction techniques, the proper orthogonal decomposition (POD) based approach is one of the most popular methods. Additionally, the POD-ROM that commonly approximates a high-order full system to a low-dimensional ROM uses proper orthogonal mode (POM) as a transformation matrix projecting the system matrices to the reduced-order space in an intrusive manner. The projection-based intrusive ROM has been applied in many research fields due to its advantages over other ROMs: parabolic equations [14], oceanic models [15,16], nonstationary Navier–Stokes equations [17], bifurcation problem [18], and linear dynamical systems [19,20]. However, most intrusive ROMs

require system matrices that are derived from the governing equation and change depending on discretization methods. Furthermore, reproducing algorithms and developing source codes are essential for implementing the ROM to various scientific and engineering problems. Such a cumbersome process is one of the major limitations of the intrusive ROM from the viewpoint of scalability, even though the intrusive ROMs have several advantages, such as showing higher accuracy and containing more flexibility than other surrogates for handling the changes in parameters.

On the other hand, the non-intrusive reduction methods commonly construct a model surrogate using input/output data without intruding into the governing equation. In general, the data set for the non-intrusive ROM is obtained from simulations and experiments, which means that simulation results from open-source or commercial software can also be used for constructing ROM. The input/output relation is approximated by post-processing the data, and this does not require a large amount of work compared with those using intrusive ROMs.

Among various non-intrusive ROMs based on the POD method, adapting the radial basis function (RBF) and discrete empirical interpolation method (DEIM) to derive the ROM was proposed by Klie [21]. A non-intrusive ROM related to the POD method was proposed by Guénot et al. [22] and Casenave et al. [23]. The non-intrusive POD-ROM for aerodynamic shape optimizations was proposed by Iliano and Quagliarella [24]. Peherstorfer and Willcox developed a non-intrusive model reduction approach for operator inference within a data-driven framework [25]. Xiao et al. proposed various non-intrusive POD-based ROM methods for Navier–Stokes equations [26,27] or fluid–structure interactions (FSIs) [28]. The non-intrusive POD/RBF method efficiently reduces the computational cost for nonlinear fluid problems [27]. However, such methods usually reconstruct and solve a new ROM when the input condition changes. Recently, non-intrusive ROM using POD/RBF was proposed by Nguyen and Kim [29] showing efficient approximation with respect to the change in parameters for nonlinear contact problems. In addition, ROMs based on POD-ANN [30], deep learning [31], and the convolutional autoencoder and predictive adversarial network approach [32] were actively proposed, widening the scope and applications of the ROM to complex fluid problems. Additionally, many studies have been performed regarding adaptive ROMs to the change of input parameter, which are usually referred to as parametric ROM (PROM). For various physical systems, reduced-basis methods [33,34], interpolation-based ROM [35–37], and PROM combined with substructuring schemes [38–40] have been developed and applied to many engineering applications. Nevertheless, research on the non-intrusive ROM that adapts to the parametric variation for dynamic problems is less active than that on the intrusive ROM.

In this study, we propose a two-step, non-intrusive ROM based on a data reconstruction process. In a fluid model, if an inlet velocity changes, several output parameters of solution response, such as the frequency, phase, and amplitude, also change nonlinearly. Because of this issue, it is difficult to construct a ROM that can analyze the problems for various initial input conditions using conventional non-intrusive methods. However, the proposed method can efficiently analyze the fluid problems for various inlet velocities without querying the governing equations. To construct such a surrogate model, the proposed approach uses the discrete Fourier transform, in particular, the fast Fourier transform (FFT), and the POD/RBF interpolation. A three-stage procedure is proposed, which consists of two-step offline stages: Step 1 and Step 2, and one online stage.

Step 1 consists of sampling analyses and adjusting data sets in the virtual time and frequency domains. The sampling analyses are performed to construct data sets with the solution responses and initial input conditions using available software packages. The study sets the frequency value that has the most dominant effect in each sampling case as a scaling factor and creates a virtual time/frequency domain using the scaling factor and interpolations. Step 1 uses the interpolation method and the scaling factor to adjust the sampled solution responses so that the peaks of each frequency response are matched at the same degree of freedom (DOF) in the virtual domain. During the reconstruction

process, the proposed approach invokes a POD/RBF interpolation method to efficiently interpolate the high-dimensional solution responses.

In Step 2, the adjusted sampled data sets are used to construct the surrogate ROM. Step 2 gathers the adjusted solution responses for the same DOF in all cases and transforms them to the virtual frequency domain using the FFT operation. The new data sets for each DOF that consist of real and imaginary numbers are constructed in the virtual frequency domain. The proposed approach constructs a surrogate ROM with interpolation formulas for three types of data sets: (1) scaling factor, (2) real numbers of each DOF, and (3) imaginary numbers of each DOF. The input variable of the interpolation formulas is the initial inlet velocity that basically changes the Reynolds number, and the POD/RBF interpolation method with respect to the parameter is used to increase efficiency.

In the online stage, the surrogate ROM approximates the data sets, such as the solution responses in the virtual frequency domain and the scaling factor for the perturbed initial inlet velocity. Because the approximated responses still exist in the virtual frequency domain, they should be returned to the original time domain. Such a process is performed in the reverse of the adjustment procedure of Step 1. Subsequently, solution responses for the perturbed input condition can be efficiently obtained.

The structure of this paper is as follows. The problem setup, which includes the governing equations of fluids and the simulation software, is introduced in Section 2. In Section 3, the data reconstruction procedures and ROM derivations are presented, providing the FFT operations and the POD/RBF interpolations. Two numerical examples, the flows past a circular cylinder and the flow around an airfoil, are illustrated in Section 4, verifying the efficiency and accuracy of the proposed approach. Finally, the conclusions are given in Section 5.

## 2. Problem Setup

In Section 2.1, the governing equations of the fluid problem are presented. Because the proposed method basically handles the solution data of numerical simulations, the software and its characteristics on the solvers and algorithm are briefly introduced.

### 2.1. Governing Equations

This study considers a two-dimensional, unsteady incompressible Newtonian fluid problem. To perform the analysis, we need to solve the governing equations that consist of the continuity equation and the Navier–Stokes equation, as expressed below:

$$\nabla \cdot \mathbf{u} = 0, \quad (1)$$

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

where  $\mathbf{u}(t; \mu) = (u_x(t; \mu), u_y(t; \mu), u_z(t; \mu))^T$  is a velocity vector at time  $t$  in a Euclidean space. Also,  $\rho$ ,  $p$ , and  $\nu$  are the density, pressure, and kinematic viscosity, respectively. A parameter  $\mu$  is introduced to express the dependency of the velocity field on the changes of the initial condition. The governing equations are solved using well-known numerical techniques during the sampling process. Thus, a discretized velocity vector represented by  $\mathbf{u}(t; \mu) = (u_1(t; \mu), u_2(t; \mu), \dots, u_N(t; \mu))^T$  is introduced, where  $N$  denotes the size of discrete velocity vector of a full-order model (FOM). For a two-dimensional flow, two times of the number of vertex (node) equals  $N$ .

### 2.2. Simulation Tool-Based Approach: OpenFOAM Solver and Algorithm

Basically, the proposed approach utilizes simulation software due to its convenience and robustness for acquiring simulation data. In this study, we adapt OpenFOAM software, which is one of the popular packages for computational fluid dynamics (CFD) simulations. Of course, the proposed method can be applied to other packages, including open-source and commercial software as well as in-house codes.

OpenFOAM provides the pressure implicit splitting operator (PISO), semi-implicit method for pressure-linked equations (SIMPLE), and PISO–SIMPLE (PIMPLE) algorithms as the main algorithms to analyze the Navier–Stokes equations. `simpleFoam` is a steady-state solver consisting of the SIMPLE algorithm to analyze the incompressible and turbulent flow problems. `pisoFoam` is a transient solver for the incompressible non-Newtonian turbulent flow problems based on the PISO algorithm. It can set the turbulence model by selecting the laminar, Reynolds-Averaged Navier–Stokes (RANS), or large eddy simulation (LES) options. Lastly, `pimpleFoam` is a large time-step transient solver for the incompressible turbulent flow that uses the PIMPLE algorithm, the merged form of the PISO and SIMPLE algorithms. `pimpleFoam` can substitute the `simpleFoam` and `pisoFoam` solvers for unsteady, incompressible flows. The PISO and SIMPLE algorithms are comprehensively explained in [41].

In this study, we analyze the pressure–velocity coupling fluid model using the `pimpleFoam` solver because it is more robust and efficient than the `pisoFoam` and `simpleFoam` solvers. The `pimpleFoam` solver is described in [42,43]. To analyze the problems of turbulent fluid, it is necessary to set an appropriate turbulence model. In this study, we set the simulation type of the turbulence model to the k-epsilon RANS model.

### 3. Data Reconstruction-Based Two-Step Non-Intrusive ROM

To efficiently compute the solution response for the perturbed initial inlet velocity, we present the data reconstruction-based non-intrusive ROM method. The proposed approach consists of two offline steps, which are data reconstruction in a time domain and a ROM construction with an interpolation model in a frequency domain. In the online stage, the constructed ROM can efficiently compute the solution responses for arbitrary initial inlet velocity. At the end of this section, the overall algorithm of the proposed method is presented.

#### 3.1. Offline Step 1: Data Reconstruction in a Virtual Time Domain

In turbulent fluid problems, even if the input condition changes linearly, the velocity profiles at the same DOF show nonlinearities with respect to the input condition. In other words, the velocity profiles at the same DOF exhibit different vibrational occurrences, amplitudes, and frequencies depending on the initial inlet velocity. Therefore, we should reconstruct the data through time-step scaling to align the frequency peaks of the solution response of all sampling cases. In the reconstruction procedure, while maintaining the magnitude of the data, the time step is scaled by adopting a scaling factor. The scaling factor of each case is respectively set, and it reconstructs the data by adjusting the scale of time steps with the scaling factor.

Firstly, a set of parameters representing the sampling points is expressed as follows:

$$S = \left\{ \mu^{(k)} \in \mathbb{R} : k = 1, 2, \dots, p \right\}, \quad (3)$$

where the superscript  $k$  denotes the sample number of inlet conditions. The sampled data with a velocity response for several initial inlet conditions can be expressed as follows:

$$\mathbf{u}^{(k)}(t) = \mathbf{u}(t; \mu^{(k)}). \quad (4)$$

The proposed approach searches for the most dominant frequency value that can be representative of each case, which has the most dominant effect on the solution response. The most dominant frequency value becomes the scaling factor for each sampling case. Thus, the data are converted into the frequency domain using the FFT operation to determine the scaling factor for each case, such that

$$\mathcal{F}\left\{ \mathbf{u}^{(k)}(t) \right\} = \mathcal{U}^{(k)}(\omega), \quad (5)$$

where  $\mathcal{F}$  denotes a Fourier transform operator. The frequency of the 1st peak point except 0 Hz is commonly the dominant frequency value in most DOFs. Therefore, the scaling factor of each case is determined with the most dominant frequency value. Let the scaling factor of sample  $k$  be  $\bar{\omega}^{(k)} = \bar{\omega}(\mu^{(k)})$ , which can be written by

$$\bar{\omega}^{(k)} = \arg \max_{\omega} |\mathcal{U}^{(k)}(\omega)|. \quad (6)$$

In Equation (6), the upper bar  $\bar{\bullet}$  is used to represent a scaling factor, which is a function of the parameter,  $\mu$ . Note that for a specific input parameter, the scaling factor is a constant. Then the time variable of each case can be scaled as follows:

$$\tilde{t}^{(k)} = \bar{\omega}^{(k)} t. \quad (7)$$

Time steps can also be scaled using  $\bar{\omega}^{(k)}$  as the same manner in Equation (7). Based on the scaling, the solution field in a scaled time can be expressed using the  $\tilde{\bullet}$  symbol, and the following relationship holds for all sample cases:

$$\tilde{\mathbf{u}}^{(k)}(\tilde{t}^{(k)}) = \mathbf{u}^{(k)}(t). \quad (8)$$

After the time scaling, each solution must be aligned in a virtual, common time domain,  $\tilde{t}$  that is defined by the following index representing maximum frequency among each case:

$$\tilde{t} := \tilde{t}^{(m)}, \quad (9)$$

where the superscript  $m$  is determined by

$$m = \arg \max_k \omega^{(k)}. \quad (10)$$

One unavoidable issue of the scaling is that the time steps of each sample case are not the same as others. To be specific, let the subscript  $i$  be the  $i$ th time step, then  $\tilde{t}_i \neq \tilde{t}_i^{(k)}$  except for  $k = m$ . Thus, since  $\tilde{\mathbf{u}}^{(k)}$  are not evaluated at identical time instances, we need to compute  $\tilde{\mathbf{u}}^{(k)}(\tilde{t}_i)$ , which requires high-fidelity simulations with sample cases once more. To avoid such cost-consuming work, we perform the interpolation of existing data with respect to the virtual time domain. However, the interpolation requires a great amount of computational cost as the solution lies in  $N$ -dimensional space. To overcome this issue, the RBF-based POD interpolation method is introduced to enhance computational efficiency. By the POD of the sampling data, proper orthogonal modes (POM),  $\Phi^{(k)} \in \mathbb{R}^{N \times R}$ , and their coefficients,  $\tilde{\mathbf{p}}^{(k)} \in \mathbb{R}^{N \times R}$ , are obtained, and the linear combination gives the approximation such that

$$\tilde{\mathbf{u}}^{(k)}(\tilde{t}_i) \approx \Phi^{(k)} \tilde{\mathbf{p}}^{(k)}(\tilde{t}_i), \quad (11)$$

where  $R$  is the dimension of the reduced system, which is usually much smaller than  $N$ . In general,  $R$  is determined by the truncation from the distribution of singular values. The interpolation is expressed using the RBF as follows:

$$\tilde{\mathbf{u}}_q^{(k)}(\tilde{t}) \approx \sum_{j=1}^R \Phi_{qj}^{(k)} \tilde{p}_j^{(k)}(\tilde{t}) = \sum_{j=1}^R \left\{ \Phi_{qj}^{(k)} \sum_{i=1}^{N_i} w_{ij}^{(k)} \phi(r(\tilde{t}, \tilde{t}_i^{(k)})) \right\}, \quad (12)$$

where  $r(\tilde{t}, \tilde{t}_i^{(k)})$  represents the Euclidean distance between the  $\tilde{t}$  and  $\tilde{t}_i^{(k)}$ , and  $\phi$  is a basis function. In this study, a multiquadric  $\phi(r) = \sqrt{1 + (r/\varepsilon)^2}$  was used as the RBF, and the weightings are obtained by using the sample data set that consists of  $\tilde{t}_i^{(k)}$  and  $\tilde{\mathbf{u}}^{(k)}(\tilde{t}_i)$ .

Substituting the sample data set into Equation (12) results in a linear system, and  $R \times N_t$  unknown weightings are determined by solving the linear system. Note that the interpolation in Equation (12) is for the reconstruction of data in a virtual time domain, which needs to be discriminated the interpolation with respect to the parametric approximation that will be shown in Section 3.2.

### 3.2. Offline Step 2: Constructing Reduced-Order Model in a Frequency Domain

In Step 2, the surrogate reduction system for approximating the scaling factor and adjusting solution responses with respect to the perturbed initial inlet velocity is constructed. To this end, new data sets from the adjusted solutions are generated to capture the dynamics of the system. By converting the adjusted responses  $\tilde{\mathbf{u}}^{(k)}(\tilde{t})$  to the virtual frequency domain using the FFT operation, new data sets are expressed as follows:

$$\mathcal{F}\left\{\tilde{\mathbf{u}}^{(k)}(\tilde{t})\right\} = \tilde{\mathcal{U}}^{(k)}(\tilde{\omega}) = \tilde{\mathcal{A}}^{(k)}(\tilde{\omega}) + i\tilde{\mathcal{B}}^{(k)}(\tilde{\omega}), \quad (13)$$

where  $\tilde{\mathcal{A}}^{(k)}(\tilde{\omega}) \in \mathbb{R}^N$  and  $\tilde{\mathcal{B}}^{(k)}(\tilde{\omega}) \in \mathbb{R}^N$  denote the new data sets for real and imaginary numbers, respectively. For a new initial inlet velocity denoted by  $\mu^*$ , we interpolate the data in the virtual frequency domain with respect to the parameter. However, even though the interpolation with respect to the new input parameter is different from that with data reconstruction in the virtual time domain, there also exists a chance to relieve computational burdens using the POD/RBF interpolation. Thus, by using the POM of the new data sets of the real and imaginary parts, the transformations can be expressed as follows:

$$\tilde{\mathcal{A}}^*(\tilde{\omega}) \approx \Phi_{\mathcal{A}}(\tilde{\omega})\tilde{\alpha}^*(\tilde{\omega}), \quad (14)$$

$$\tilde{\mathcal{B}}^*(\tilde{\omega}) \approx \Phi_{\mathcal{B}}(\tilde{\omega})\tilde{\beta}^*(\tilde{\omega}), \quad (15)$$

where each POM is obtained using  $N \times p$  vectors of  $\tilde{\mathcal{A}}^{(k)}$  and  $\tilde{\mathcal{B}}^{(k)}$  at the discrete frequency,  $\tilde{\omega}$ , respectively. Corresponding POD/RBF interpolation of the component of  $\tilde{\mathcal{A}}^*$  and  $\tilde{\mathcal{B}}^*$  are written as

$$\tilde{\mathcal{A}}_i^*(\tilde{\omega}) \approx \sum_{j=1}^R \Phi_{\mathcal{A},ij}(\tilde{\omega})\tilde{\alpha}_j^*(\tilde{\omega}) = \sum_{j=1}^R \left\{ \Phi_{\mathcal{A},ij}(\tilde{\omega}) \sum_{k=1}^p w_{\mathcal{A},jk} \phi\left(r\left(\mu^*, \mu^{(k)}\right)\right) \right\}, \quad (16)$$

$$\tilde{\mathcal{B}}_i^*(\tilde{\omega}) \approx \sum_{j=1}^R \Phi_{\mathcal{B},ij}(\tilde{\omega})\tilde{\beta}_j^*(\tilde{\omega}) = \sum_{j=1}^R \left\{ \Phi_{\mathcal{B},ij}(\tilde{\omega}) \sum_{k=1}^p w_{\mathcal{B},jk} \phi\left(r\left(\mu^*, \mu^{(k)}\right)\right) \right\}, \quad (17)$$

where  $R \ll N$ .  $w_{\mathcal{A},jk}$  and  $w_{\mathcal{B},jk}$  represent  $(j, k)$  component of weighting coefficient,  $w_{\mathcal{A}}$  and  $w_{\mathcal{B}}$ , which are obtained by solving conventional RBF linear maps between the function values and parameters constructed by the sampling points. Also,  $\Phi_{\mathcal{A},ij}$  and  $\Phi_{\mathcal{B},ij}$  are  $(i, j)$  component of the POM,  $\Phi_{\mathcal{A}}$  and  $\Phi_{\mathcal{B}}$ . As stated at the end of Section 3.1, the interpolations in Equations (16) and (17) are for a new parametric input, which needs to be distinguished from the interpolation for the data reconstruction of the offline step 1.

The POD/RBF interpolation models for each DOF are constructed using the POMs, weight coefficients, and radial basis function. Because the solution response obtained using Equations (14)–(17) stays in the virtual domain, it should be transformed to the original time domain. Therefore, the scaling factor is also required for the perturbed initial inlet velocity. The interpolation model for the scaling factor is constructed using the set of scaling factors and the RBF interpolations as follows:

$$\bar{\omega}^* = \bar{\omega}(\mu^*) = \sum_{k=1}^p w_{\omega,k} \phi\left(r\left(\mu^*, \mu^{(k)}\right)\right), \quad (18)$$

where the weighting,  $w_\omega \in \mathbb{R}^p$ , is determined by solving a linear system substituting  $\mu^{(k)}$  into  $\mu^*$  of Equation (18).

Finally, the surrogate reduction system is completed using three types of interpolation formulas about the (1) scaling factor, (2) real, and (3) imaginary numbers in the virtual frequency domain.

### 3.3. Online Stage

In the online stage, the solution response is approximated for the perturbed input parameter using the surrogate ROM. First, the interpolations of Equations (16)–(18) are performed with respect to a new input  $\mu^* \notin S$ . Since the POM and the weighting coefficients are computed in the offline stage, the results associated with a new input are efficiently approximated via additions and multiplications. After recovering the parts of real and imaginary data in Equations (16) and (17), the inverse FFT (IFFT) operation is applied to convert them to the virtual time domain such that

$$\mathcal{F}^{-1}\{\tilde{\mathcal{A}}^*(\tilde{\omega}) + i\tilde{\mathcal{B}}^*(\tilde{\omega})\} = \tilde{\mathbf{u}}^*(\tilde{t}), \quad (19)$$

As  $\tilde{\mathbf{u}}^*$  exists in the virtual time domain, it should be transformed to the original time using the reverse procedure of the offline stage. To return the virtual domain to the original one, the scaling factor for the target input parameter is approximated via the surrogate reduction system, and the virtual time scale is divided by the factor, as shown in Equation (7). Finally, the solution response of the new input parameter can be obtained, which can be written as

$$\mathbf{u}^*(t) = \tilde{\mathbf{u}}^*(\tilde{t}). \quad (20)$$

The surrogate ROM efficiently computes the solution response for various input parameters in the online stage. Algorithm 1 shows the computational procedure of the proposed method. Step 1 and Step 2 are distinguished depending on the data before and after the reconstruction process. The performances of the two-step non-intrusive ROM method are demonstrated by the numerical problems in the next section.

---

#### Algorithm 1: Data reconstruction-based two-step non-intrusive ROM

---

##### //Offline stage—Step 1//

- 1: **for**  $k = 1, 2, \dots, p$ , **do**
- 2: Solve FOM at sample parameters,  $\mu^{(k)}$ .
- 3: Execute FFT of the sampled data sets and compute scaling factors,  $\bar{\omega}^{(k)}$ .
- 4: **end for**
- 5: Find a virtual time domain,  $\tilde{t} \leftarrow \tilde{t}^{(m)}$  as in Equation (10).
- 6: **for**  $k = 1, 2, \dots, p$ , **do**
- 7: Adjust data sets to the virtual time domain.
- 8: Apply POD to the adjusted data.
- 9: Perform RBF interpolation in a reduced dimension for data reconstruction.

##### //Offline stage—Step 2//

- 10: Execute FFT of the reconstructed data.
- 11: Apply POD to the reconstructed data.
- 12: **end for**

##### //Online stage//

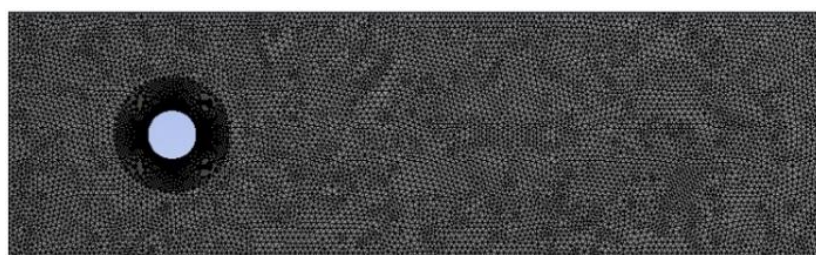
- 13: Perform RBF interpolations to find  $\omega^*$ ,  $\tilde{\alpha}^*$  and  $\tilde{\beta}^*$  of a new input parameter,  $\mu^*$ .
  - 14: Compute  $\tilde{\mathcal{A}}^*$  and  $\tilde{\mathcal{B}}^*$  by the recovering process of Equations (14) and (15).
  - 15: Execute IFFT to find  $\tilde{\mathbf{u}}^*(\tilde{t})$ .
  - 16: Perform scaling using  $\bar{\omega}^*$  to return to the original time domain.
-

#### 4. Numerical Examples

In this section, the proposed method is applied to two examples of Navier–Stokes problems. The examples are modeled with a 2-D turbulent fluid with an unsteady incompressible Newtonian fluid. In the problems, the initial inlet velocity is set as the input parameter changing the Reynolds number for each problem. Depending on the change in the initial inlet velocity, the response does not change drastically. However, the frequency, amplitude, and phase change nonlinearly. As stated in Section 2.2, sampling analyses are performed using the pimpleFoam solver of the OpenFOAM software to construct a surrogate ROM for the problems.

##### 4.1. Flow Past a Circular Cylinder in Channel

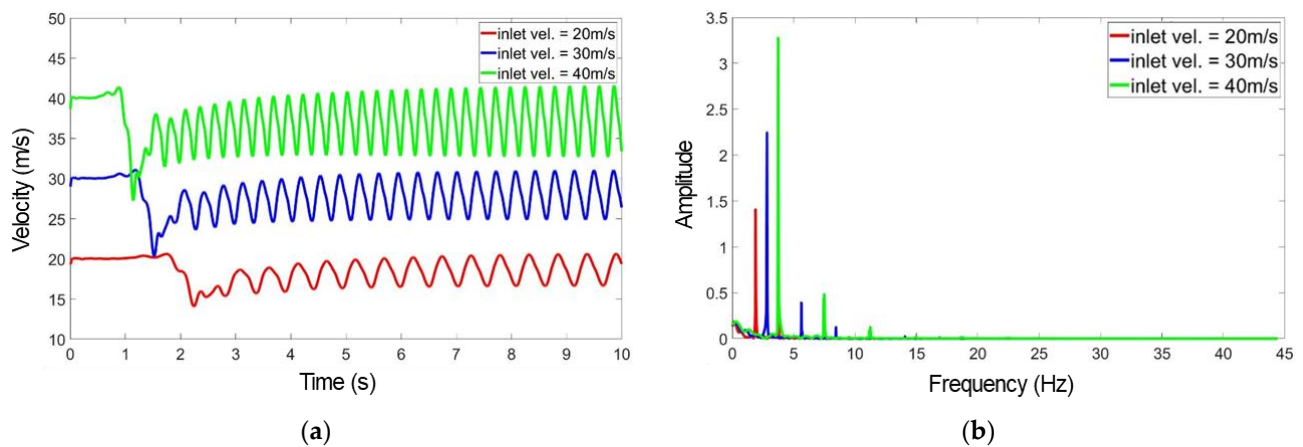
The two-dimensional circular cylinder in the channel problem is investigated to validate the efficiency and accuracy of the surrogate ROM. The problem is a model with dimensions  $50 \times 15 \text{ m}^2$  and a cylinder with a radius of 1.5 m. The model is meshed with 13,424 nodes and 21,526 triangular elements, as shown in Figure 1. The left and right edges are set as the entrance and exit boundary conditions, respectively, and the Dirichlet boundary condition is applied to the upper and lower walls and around the cylinder. The temperature of the fluid is set to  $20^\circ\text{C}$ , and the kinematic viscosity value is  $1.5 \times 10^{-5} \text{ m}^2/\text{s}$ .



**Figure 1.** Example 1: Mesh of the flows past a circular cylinder in the channel.

The sampling analyses are simulated at inlet velocities of 20 m/s, 30 m/s, and 40 m/s using OpenFOAM. Thus, the x-directional velocity of the left edge in Figure 1 was imposed as a constant value with respect to time. Figure 2 represents time-dependent velocities of each inlet condition, and as expected, initiations of oscillation, amplitudes, and time periods, also presented in Figure 2b, are different from each other. For each sampling case, Table 1 represents the analysis time, CPU time, and the scaling factors. As shown in Table 1, the sampling analysis of each case was performed for the analysis time that is inversely proportional to the scaling factor value. This is related to the time step changed using the scaling factor in the Step 1 procedure. The time domain of the sampled data is adjusted, and the data are reconstructed, as presented in Figure 3. If the sampling analyses are performed during the same analysis time, garbage data is generated. Therefore, because the scaling factors are proportional to the inlet velocity, we run the sampling analysis for a suitable analysis time based on the inlet velocity.

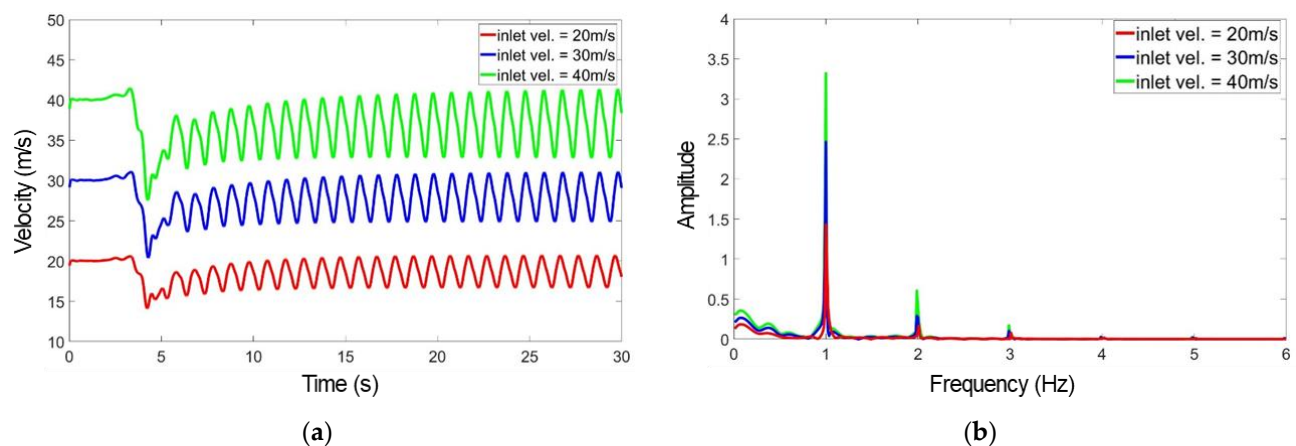
The validation case is performed for the initial inlet velocity of 35 m/s to verify the performance of the ROM. The CPU time and solution response of the OpenFOAM are used to compare computational efficiency and accuracy. Using the offline stage of the two-step non-intrusive ROM method, we can construct the surrogate reduction model for this problem. The scaling factor is computed to 3.2778 via the surrogate ROM, and the solution response is obtained for 0.006–25.93 s with a time step of 0.0024 s. OpenFOAM spent a CPU time of 2.024 s, whereas the reduction system took a CPU time of only 41.59 s. Consequently, the proposed method is 48.67 times more efficient than OpenFOAM. In fact, the CPU time of the ROM majorly contains the RBF interpolation on the parameter and IFFT for the scaling back to the time domain, which is similar to the costs of the RBF interpolation with respect to the virtual time steps and FFT. Additionally, the cost of  $2p$  times POD is added to the total offline computations, as presented in Algorithm 1.



**Figure 2.** Example 1: Solution responses of sampling cases in x-direction: (a) Time domain; (b) Frequency domain.

**Table 1.** Example 1: Sampling cases.

Inlet Velocity (m/s)	$[t_0, t_f]$ (s)	$\omega^{(k)}$	CPU Time (s)
20	[0, 45]	1.889	3208
30	[0, 35]	2.822	2612
40	[0, 30]	3.733	2338

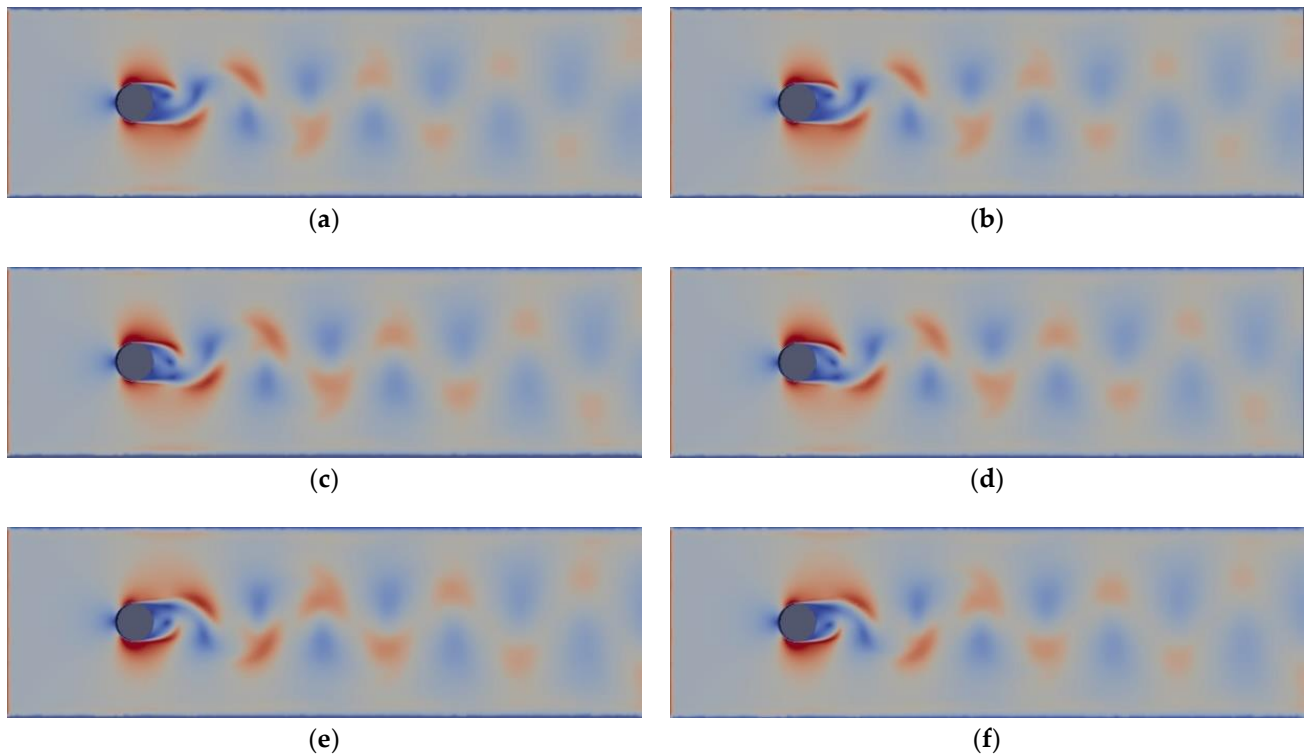


**Figure 3.** Example 1: Adjusted solution responses in x-direction: (a) Virtual time domain; (b) Virtual frequency domains.

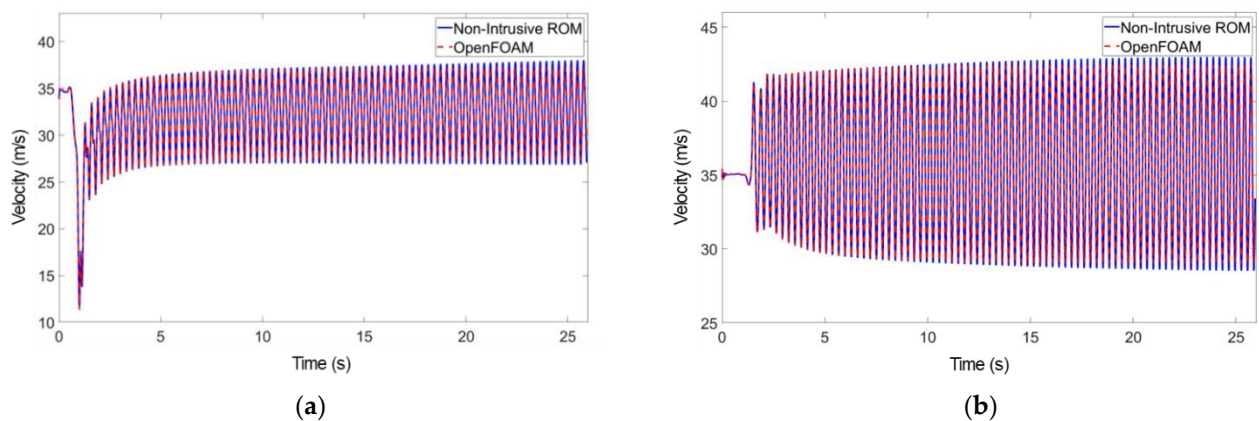
Figure 4 illustrates the velocity distribution results of the proposed method and OpenFOAM at specific times, 4.0016 s, 12.002 s, and 24.0712 s. Figure 5 shows the x-direction velocity profiles at locations (20.9761, 7.8896) and (39.3667, 11.3086). The blue and red lines indicate the results of the proposed ROM method and OpenFOAM, respectively. It can be observed that the reduction system is in good agreement with the OpenFOAM solution. Figure 6 shows the velocity profiles for different inlet velocities obtained by using the proposed method at the same locations as Figure 5. The initiations of the oscillation and the magnitudes change nonlinearly with respect to the inlet condition. Furthermore, the relative root mean square error (RRMSE) is used to estimate the accuracy of the proposed method. The RRMSE method can measure the accuracy of the high-dimensional data at each time step. It can be expressed as follows:

$$(RRMSE) = \frac{\sqrt{\sum_{i=1}^N (u_i^*(t) - \bar{u}_i^*(t))^2}}{\sqrt{\sum_{i=1}^N (\bar{u}_i^*(t))^2}}, \quad (21)$$

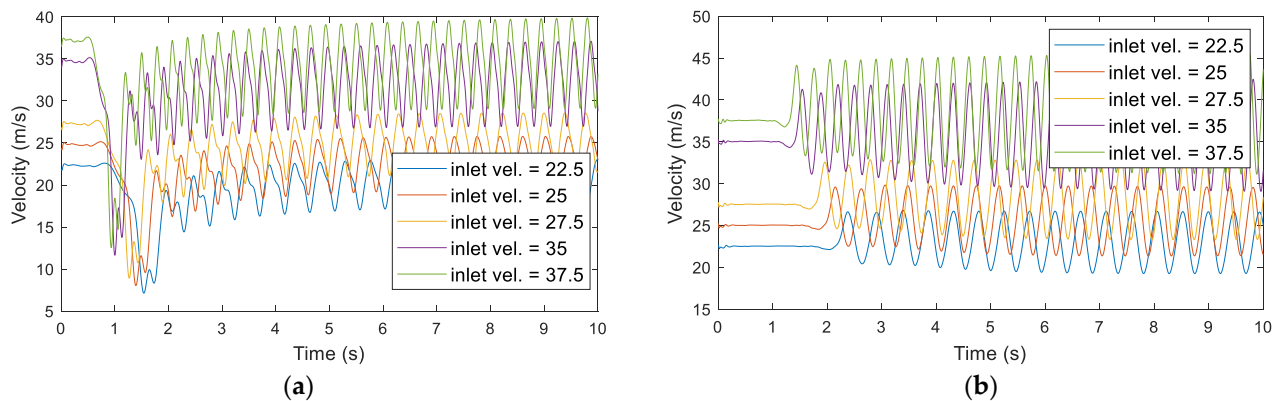
where the upper bar represents the solution obtained by the OpenFOAM and  $u_i^*$  is the velocity computed by the proposed ROM. The RRMSEs of the velocity in x-direction at each time step are depicted in Figure 7. Although the error increases with oscillations that can be understood as a basic characteristic of the interpolation-based ROMs [26–28,32–37], the maximum error is relatively small, with a value of 0.0515.



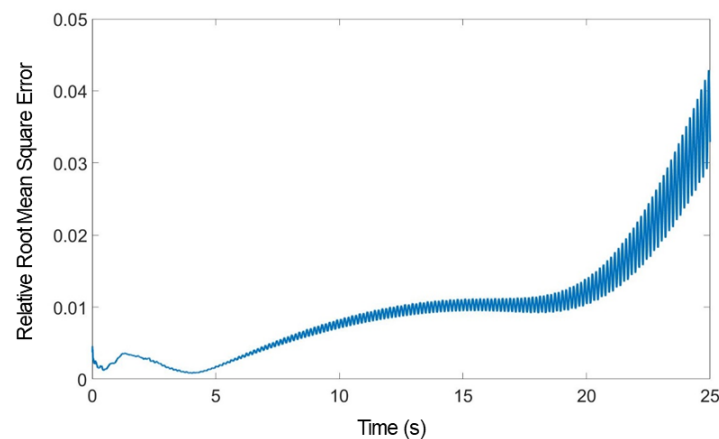
**Figure 4.** Example 1: Adjusted solution responses in x-direction with an inlet velocity of 35 m/s: (a) ROM at  $t = 4.0016$  s; (b) OpenFOAM at  $t = 4.0016$  s; (c) ROM at  $t = 12.0002$  s; (d) OpenFOAM at  $t = 12.0002$  s; (e) ROM at  $t = 24.0712$  s; (f) OpenFOAM at  $t = 24.0712$  s.



**Figure 5.** Example 1: x-direction velocity profiles for two-step non-intrusive ROM and OpenFOAM at two random locations: (a) Location (20.9761, 7.8896); (b) Location (39.3667, 11.3086).



**Figure 6.** Example 1: x-direction velocity profiles for different inlet velocities: (a) Location (20.9761, 7.8896); (b) Location (39.3667, 11.3086).



**Figure 7.** Example 1: RRMSEs of the proposed method from OpenFOAM.

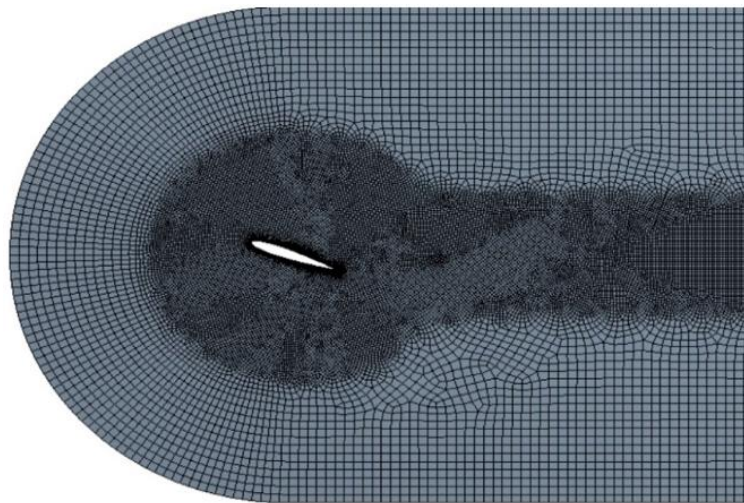
#### 4.2. Flow around Airfoil

The second example is a two-dimensional turbulence fluid problem with flow around an airfoil using the NACA0012 model. The width and height of the full model are 9 m and 6 m, respectively. The center of the airfoil is located at (3.5, 3), and the angle of attack is set to  $\alpha = 20^\circ$  to generate a wake phenomenon. The model is meshed with 21,955 rectangular elements, as illustrated in Figure 8. At the edge of the airfoil, the Dirichlet boundary condition is applied, while the front-facing semicircle and other edges are set to the inlet and outlet boundary conditions, respectively. The temperature of the fluid is set to 20 °C, and the kinematic viscosity is  $1.5 \times 10^{-5} \text{ m}^2/\text{s}$ .

The sampling analyses are simulated at inlet velocities of 130 m/s, 150 m/s, and 170 m/s using the OpenFOAM. The solution responses for all cases are collected for 0–20 s analysis time with the time step of 0.002 s. Unlike the first example, because the ratio of inlet velocity is closed to 1, the sampling analyses are simulated with the same analysis time. To demonstrate the efficiency and accuracy of the proposed method, the analysis is performed for the inlet velocity of 140 m/s using the OpenFOAM and surrogate reduction system.

As presented in Table 2, the scaling factors obtained using the determination procedure are 84.65, 97.8, and 110.9. It can be observed that the scaling factor is calculated using a rate similar to the inlet velocity. Next, the surrogate reduction system for flow around an airfoil problem is constructed using the scaling factor and sampled data set. The solution response is obtained for 0.0033–18.5245 s using the surrogate reduction system. Figure 9 illustrates the velocity distributions of the proposed method and OpenFOAM at specific times. In addition, Figure 10 plots the x-direction velocity profile for the specific positions: (4.1175, 2.9034) and (7.1536, 1.9996). The results of the reduction system are in good agreement with OpenFOAM. To estimate the accuracy, the RRMSE method is used, and the error values

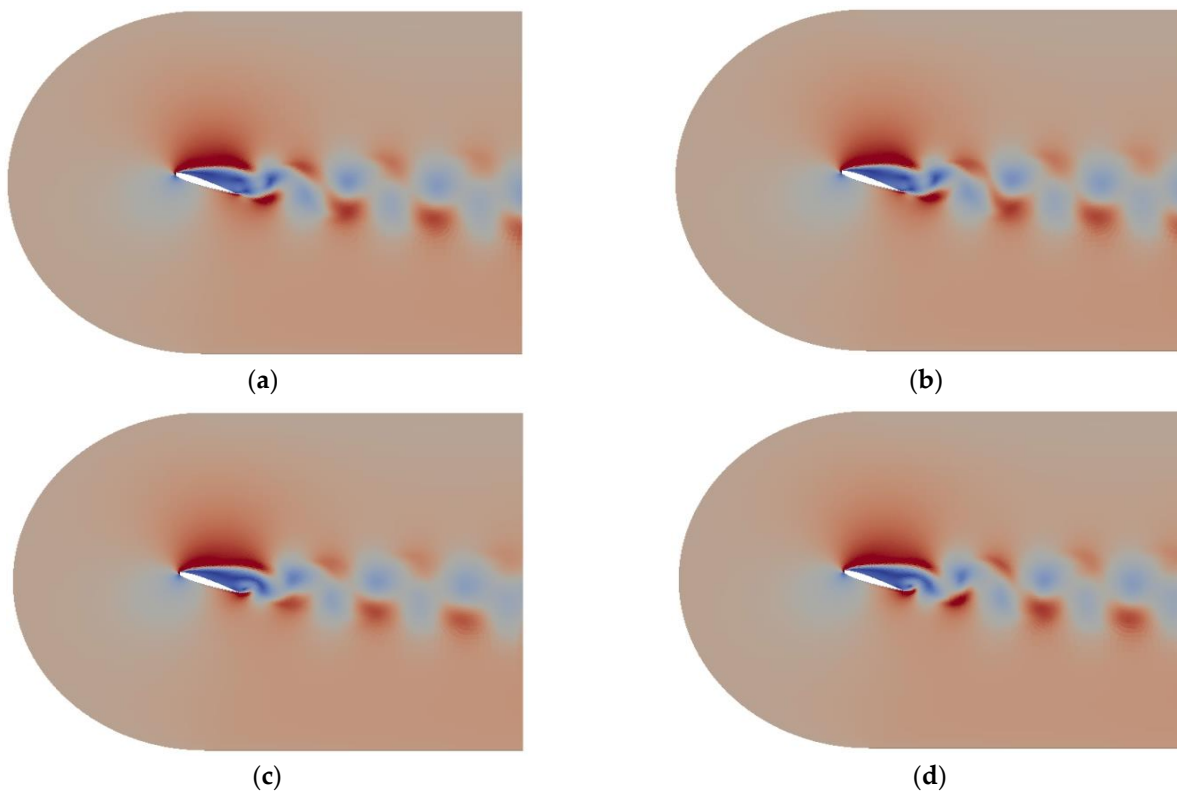
of each time step are depicted in Figure 11. Although the RRMSE increases gradually, the maximum error value is estimated to be 0.0285.



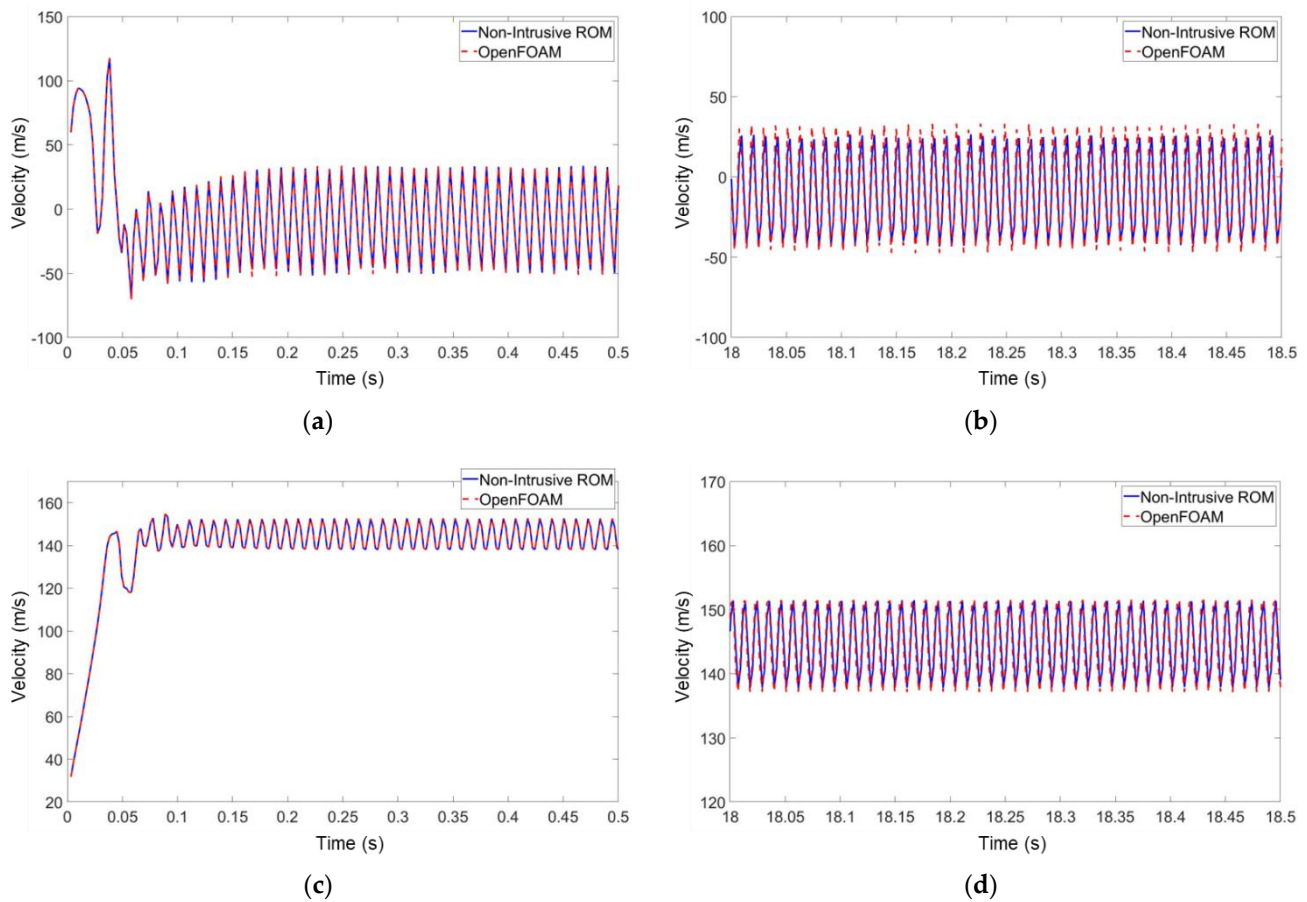
**Figure 8.** Example 2: Mesh of flow around an airfoil (NACA0012).

**Table 2.** Example 2: Sampling cases.

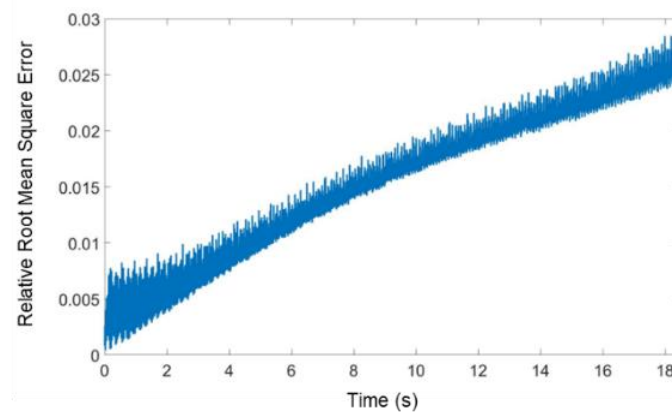
Inlet Velocity (m/s)	$[t_0, t_f]$ (s)	$\omega^{(k)}$	CPU Time (s)
130	[0, 20]	84.65	41,780
150	[0, 20]	97.80	48,473
170	[0, 20]	110.9	55,223



**Figure 9.** Example 2: Comparison of velocity distributions of the proposed ROM and OpenFOAM with an inlet velocity of 140 m/s: (a) ROM at  $t = 1.5248$  s; (b) OpenFOAM at  $t = 1.5248$  s; (c) ROM at  $t = 17.1062$ s; (d) OpenFOAM at  $t = 17.1062$ s.



**Figure 10.** Example 2: Comparison of x-directional velocity profiles of the proposed ROM and OpenFOAM at several locations: (a) 0–0.5 s at (4.1175, 2.9034); (b) 18–18.5 s at (4.1175, 2.9034); (c) 0–0.5 s at (7.1536, 1.9996); (d) 18–18.5 s at (7.1536, 1.9996).



**Figure 11.** Example 2: RRMSEs of the proposed method from OpenFOAM.

Finally, we measure the efficiency of the proposed method using the CPU times. The OpenFOAM and the proposed method took 40.613 s and 41.38 s, respectively, to analyze the problem for the perturbed initial inlet velocity. Comparing the CPU times, the proposed method is 981.46 times faster than the OpenFOAM. Consequently, the proposed method is well applied to the airfoil model and shows satisfactory accuracy and high efficiency.

## 5. Conclusions

We present a two-step non-intrusive ROM method based on data reconstruction by RBF interpolation and POD-based dimensionality reduction. The conventional intrusive

reduction methods require intrusive works such as studying the governing equations or modifying the source code to construct reduction systems. However, the proposed method does not require intrusive works because the surrogate reduced order model is constructed using only the input/output data obtained from the analysis software. Instead, the proposed approach requires dealing with analysis programs and post-processing of data. The proposed approach uses the FFT operation and POD/RBF interpolation method to adjust the solution data sets and to construct the surrogate ROM with interpolation formulas. The proposed ROM can efficiently compute the solution response for various input conditions. In this study, two numerical examples of the Navier–Stokes problem are investigated to demonstrate the accuracy and efficiency of the proposed method. Consequently, the two-step non-intrusive ROM method was well applied with high efficiency and reasonable accuracy.

The sampling analyses and determination of the scaling factor in the offline process require intensive analyses in future studies. The time step of the sampling analyses influences the computation time of Step 1. If the sampling time step is smaller, the accuracy of the scaling factor increases, whereas the calculation time of Step 1 increases. Further, in the case of a problem with strong nonlinearity, because the solution data sets are complex, the determination process of the scaling factor will become important. In future investigations, the proposed method is expected to be utilized for the analysis of fluid–structure interaction and highly nonlinear fluid for various applications. Additionally, the extension to problems with multiple parameters is a promising research topic.

**Author Contributions:** Conceptualization, J.L. (Jonggeon Lee), E.K. and J.L. (Jaehun Lee); methodology, J.L. (Jonggeon Lee) and E.K.; software, J.L. (Jonggeon Lee); validation, J.L. (Jonggeon Lee) and J.L. (Jaehun Lee); formal analysis, J.L. (Jonggeon Lee); investigation, J.L. (Jonggeon Lee), E.K. and J.L. (Jaehun Lee); resources, J.L. (Jaehun Lee); data curation, J.L. (Jonggeon Lee) and J.L. (Jaehun Lee); writing—original draft preparation, J.L. (Jonggeon Lee); writing—review and editing, J.L. (Jaehun Lee); visualization, J.L. (Jonggeon Lee); supervision, J.L. (Jaehun Lee). All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the 2019 research program of Dongguk University.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Davidsson, P.; Sandberg, G. A reduction method for structure-acoustic and poroelastic-acoustic problems using interface-dependent Lanczos vectors. *Comput. Methods Appl. Mech. Eng.* **2006**, *195*, 1933–1945. [\[CrossRef\]](#)
- Hoang, K.C.; Fu, Y.; Song, J.H. An hp-proper orthogonal decomposition–moving least squares approach for molecular dynamics simulation. *Comput. Methods Appl. Mech. Eng.* **2016**, *298*, 548–575. [\[CrossRef\]](#)
- Willcox, K.; Peraire, J. Balanced model reduction via the proper orthogonal decomposition. *AIAA J.* **2002**, *40*, 2323–2330. [\[CrossRef\]](#)
- Gratton, D.; Willcox, K. Reduced-Order, Trajectory Piecewise-Linear Models for Nonlinear Computational Fluid Dynamics. In Proceedings of the 34th AIAA Fluid Dynamics Conference and Exhibit, Portland, OR, USA, 28 June 2004. [\[CrossRef\]](#)
- Walton, S.; Hassan, O.; Morgan, K. Reduced order modelling for unsteady fluid flow using proper orthogonal decomposition and radial basis functions. *Appl. Math. Model.* **2013**, *37*, 8930–8945. [\[CrossRef\]](#)
- Lassila, T.; Manzoni, A.; Quarteroni, A.; Rozza, G. Model order reduction in fluid dynamics: Challenges and perspectives. In *Reduced Order Methods for Modeling and Computational Reduction*, 1st ed.; Quarteroni, A., Rozza, G., Eds.; Springer: Cham, Switzerland, 2014; pp. 235–273. [\[CrossRef\]](#)
- Liang, Y.C.; Lin, W.Z.; Lee, H.P.; Lim, S.P.; Lee, K.H.; Sun, H. Proper orthogonal decomposition and its applications—part II: Model reduction for MEMS dynamical analysis. *J. Sound. Vib.* **2002**, *256*, 515–532. [\[CrossRef\]](#)
- Daescu, D.N.; Navon, I.M. A dual-weighted approach to order reduction in 4DVAR data assimilation. *Mon. Weather Rev.* **2008**, *136*, 1026–1041. [\[CrossRef\]](#)
- Peherstorfer, B.; Willcox, K. Dynamic data-driven reduced-order models. *Comput. Methods Appl. Mech. Eng.* **2015**, *291*, 21–41. [\[CrossRef\]](#)
- Ghavamian, F.; Tiso, P.; Simone, A. POD–DEIM model order reduction for strain-softening viscoplasticity. *Comput. Methods Appl. Mech. Eng.* **2017**, *317*, 458–479. [\[CrossRef\]](#)

11. Guo, M.; Hesthaven, J.S. Reduced order modeling for nonlinear structural analysis using gaussian process regression. *Comput. Methods Appl. Mech. Eng.* **2018**, *341*, 807–826. [\[CrossRef\]](#)
12. Lee, J.; Cho, M. Efficient design optimization strategy for structural dynamic systems using a reduced basis method combined with an equivalent static load. *Struct. Multidiscip. Optim.* **2018**, *58*, 1489–1504. [\[CrossRef\]](#)
13. Martynov, K.; Wever, U. On polynomial hyperreduction for nonlinear structural mechanics. *Int. J. Numer. Methods Eng.* **2019**, *118*, 701–717. [\[CrossRef\]](#)
14. Sun, P.; Luo, Z.; Zhou, Y. Some reduced finite difference schemes based on a proper orthogonal decomposition technique for parabolic equations. *Appl. Numer. Math.* **2010**, *60*, 154–164. [\[CrossRef\]](#)
15. Cao, Y.; Zhu, J.; Navon, I.M.; Luo, Z. A reduced-order approach to four-dimensional variational data assimilation using proper orthogonal decomposition. *Int. J. Numer. Methods Fluids* **2010**, *53*, 1571–1583. [\[CrossRef\]](#)
16. Luo, Z.; Chen, J.; Zhu, J.; Wang, R.; Navon, I.M. An optimizing reduced order FDS for the tropical Pacific Ocean reduced gravity model. *Int. J. Numer. Methods Fluids* **2007**, *55*, 143–161. [\[CrossRef\]](#)
17. Luo, Z.D.; Wang, R.W.; Zhu, J. Finite difference scheme based on proper orthogonal decomposition for the nonstationary Navier-Stokes equations. *Sci. China Ser. A Math.* **2007**, *50*, 1186–1196. [\[CrossRef\]](#)
18. Cortés, J.; Herrero, H.; Pla, F. A Galerkin/POD reduced-order model from eigenfunctions of non-converged time evolution solutions in a convection problem. *Mathematics* **2022**, *10*, 905. [\[CrossRef\]](#)
19. Kim, T. Frequency-domain Karhunen-Loève method and its application to linear dynamic systems. *AIAA J.* **1998**, *36*, 2117–2123. [\[CrossRef\]](#)
20. Kim, Y.; Wang, K.; Choi, Y. Efficient space-time reduced order model for linear dynamical systems in Python using less than 120 lines of code. *Mathematics* **2021**, *9*, 1690. [\[CrossRef\]](#)
21. Klie, H. Unlocking Fast Reservoir Predictions via Nonintrusive Reduced-Order Models. In *SPE Reservoir Simulation Symposium*; OnePetro: The Woodlands, TX, USA, 2013. [\[CrossRef\]](#)
22. Guénot, M.; Lepot, I.; Sainvitu, C.; Goblet, J.; Coelho, R.F. Adaptive sampling strategies for non-intrusive POD-based surrogates. *Eng. Comput.* **2013**, *30*, 521–547. [\[CrossRef\]](#)
23. Casenave, F.; Ern, A.; Lelièvre, T. A nonintrusive reduced basis method applied to aeroacoustic simulations. *Adv. Comput. Math.* **2015**, *41*, 961–986. [\[CrossRef\]](#)
24. Iuliano, E.; Quagliarella, D. Aerodynamic Shape Optimization via Non-Intrusive POD-Based Surrogate Modelling. In *Proceedings of the 2013 IEEE Congress on Evolutionary Computation*, Cancun, MX, USA, 20 June 2013. [\[CrossRef\]](#)
25. Peherstorfer, B.; Willcox, K. Data-driven operator inference for nonintrusive projection-based model reduction. *Comput. Methods Appl. Mech. Eng.* **2016**, *306*, 196–215. [\[CrossRef\]](#)
26. Xiao, D.; Fang, F.; Buchan, A.G.; Pain, C.C.; Navon, I.M.; Muggeridge, A. Non-intrusive reduced order modelling of the Navier-Stokes equations. *Comput. Methods Appl. Mech. Eng.* **2015**, *293*, 522–541. [\[CrossRef\]](#)
27. Xiao, D.; Fang, F.; Pain, C.; Hu, G. Non-intrusive reduced-order modelling of the Navier-Stokes equations based on RBF interpolation. *Int. J. Numer. Methods Fluids* **2015**, *79*, 580–595. [\[CrossRef\]](#)
28. Xiao, D.; Yang, P.; Fang, F.; Xiang, J.; Pain, C.C.; Navon, I.M. Non-intrusive reduced order modelling of fluid-structure interactions. *Comput. Methods Appl. Mech. Eng.* **2016**, *303*, 35–54. [\[CrossRef\]](#)
29. Nguyen, M.-N.; Kim, H.-G. An efficient PODI method for real-time simulation of indenter contact problems using RBF interpolation and contact domain decomposition. *Comput. Methods Appl. Mech. Eng.* **2022**, *388*, 114215. [\[CrossRef\]](#)
30. Fresca, S.; Manzoni, A. Real-time simulation of parameter-dependent fluid flows through deep learning-based reduced order models. *Fluids* **2021**, *6*, 259. [\[CrossRef\]](#)
31. Cheng, F.; Xu, H.; Feng, X. Model order reduction method based on (r) POD-ANNs for parameterized time-dependent partial differential equations. *Comput. Fluids* **2022**, *241*, 105481. [\[CrossRef\]](#)
32. Heaney, C.E.; Wolffs, Z.; Tómasson, J.A.; Kahouadji, L.; Salinas, P.; Nicolle, A.; Navon, I.M.; Matar, O.K.; Srinil, N.; Pain, C.C. An AI-based non-intrusive reduced-order model for extended domains applied to multiphase flow in pipes. *Phys. Fluids* **2022**, *34*, 055111. [\[CrossRef\]](#)
33. Rozza, G.; Huynh, D.B.P.; Patera, A.T. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Arch. Comput. Methods Eng.* **2008**, *15*, 229–275. [\[CrossRef\]](#)
34. Rashkov, P. Reduced basis approximation for a spatial Lotka-Volterra model. *Mathematics* **2022**, *10*, 1983. [\[CrossRef\]](#)
35. Amsallem, D.; Farhat, C. Interpolation method for adapting reduced-order models and application to aeroelasticity. *AIAA J.* **2008**, *46*, 1803–1813. [\[CrossRef\]](#)
36. Amsallem, D.; Cortial, J.; Carlberg, K.; Farhat, C. A method for interpolating on manifolds structural dynamics reduced-order models. *Int. J. Numer. Methods Eng.* **2009**, *80*, 1241–1258. [\[CrossRef\]](#)
37. Amsallem, D.; Farhat, C. An online method for interpolating linear parametric reduced-order models. *SIAM J. Sci. Comput.* **2011**, *33*, 2169–2198. [\[CrossRef\]](#)
38. Lee, J.; Cho, M. An interpolation-based parametric reduced order model combined with component mode synthesis. *Comput. Methods Appl. Mech. Eng.* **2017**, *319*, 258–286. [\[CrossRef\]](#)
39. Lee, J. A dynamic substructuring-based parametric reduced-order model considering the interpolation of free-interface substructural modes. *J. Mech. Sci. Tech.* **2018**, *32*, 5831–5838. [\[CrossRef\]](#)

- 
40. Lee, J. A parametric reduced-order model using substructural mode selections and interpolation. *Comput. Struct.* **2019**, *212*, 199–214. [[CrossRef](#)]
  41. Ferziger, J.H.; Perić, M.; Street, R.L. *Computational Methods for Fluid Dynamics*, 4th ed.; Springer: Cham, Switzerland, 2020; pp. 183–226. [[CrossRef](#)]
  42. Greenshields, C.J. *OpenFOAM Programmer's Guide*; Version 3.0.1; OpenFOAM Foundation Ltd.: London, UK, 2015.
  43. Greenshields, C.J. *OpenFOAM User Guide*; Version 10; OpenFOAM Foundation Ltd.: London, UK, 2022.