Manuscript version: Author's Accepted Manuscript

The version presented in WRAP is the author's accepted manuscript and may differ from the published version or Version of Record.

Persistent WRAP URL:

<http://wrap.warwick.ac.uk/144405>

How to cite:

Please refer to published version for the most recent bibliographic citation information. If a published version is known of, the repository item page linked to above, will contain details on accessing it.

Copyright and reuse:

The Warwick Research Archive Portal (WRAP) makes this work by researchers of the University of Warwick available open access under the following conditions.

Copyright © and all moral rights to the version of the paper presented here belong to the individual author(s) and/or other copyright owners. To the extent reasonable and practicable the material made available in WRAP has been checked for eligibility before being made available.

Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge. Provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.

Publisher's statement:

Please refer to the repository item page, publisher's statement section, for further information.

For more information, please contact the WRAP Team at: wrap@warwick.ac.uk.

Machine Learning Based Surrogate Modeling of Aerodynamic Flow around Distributed Structures

Jincheng Zhang^{*} and Xiaowei Zhao[†]
University of Warwick, Coventry, CV47AL, UK

A machine learning based surrogate modeling method for distributed fluid systems is proposed in this paper, where a dimensionality reduction technique is used to reduce the flow field dimension and a regression model is used to predict the reduced coefficients from the input parameters. The surrogate modeling method is specifically designed to tackle the fluid systems involving distributed aerodynamic structures, and its performance is illustrated by the application on the wake flow around wind turbine arrays in atmospheric boundary layer. The main idea is to first decompose the whole fluid domain into subdomains, then carry out surrogate modeling for each subdomain by treating both the boundary information and the distributed flow parameters as the input parameters, and finally obtain the whole flow field by combining the flow field of each subdomain with the consideration of the matching condition at the subdomain interface. The proposed surrogate modeling method is applied to two test cases, 1D Poisson's equation and a high-fidelity wind farm wake model. The results demonstrate the great efficiency and accuracy of the surrogate model and its excellent scalability to distributed systems of different scales.

Nomenclature

$\mathbf{a}$	=	bias term in auto-encoder
$\mathbf{b}$	=	bias term in NN regressor
d	=	distributed parameters
D	=	rotor diameter
$\mathbf{g}$	=	forward transform of dimensionality reduction
$\hat{\mathbf{g}}^{-1}$	=	inverse transform of dimensionality reduction
I	=	flow quantities at subdomain interface
K	=	number of distributed structures
l	=	1D domain length

^{*}Marie Curie Early Stage Researcher, School of Engineering, jincheng.zhang@warwick.ac.uk.

[†]Professor, School of Engineering, xiaowei.zhao@warwick.ac.uk (corresponding author).

$\mathcal{M}$	=	regression model
N	=	size of training samples
N_r	=	dimension of the reduced coefficients
N_h	=	hidden-layer neuron number
q	=	heat source term
S	=	magnitude of the heat source
T	=	temperature
T_L	=	temperature at the left boundary
T_R	=	temperature at the right boundary
$\mathcal{U}$	=	flow field by CFD simulations
$\mathcal{U}^r$	=	reduced representations of the flow field
$\hat{\mathcal{U}}$	=	flow field predicted by surrogate model
$\mathbf{v}$	=	weight matrix in auto-encoder
$\mathbf{w}$	=	weight matrix in NN regressor
$\mathbf{Z}$	=	collection of flow fields
α	=	POD coefficient
$\boldsymbol{\alpha}$	=	vector of POD coefficients
α_{NN}	=	L2 regularization coefficient in NN training
β	=	thermal diffusivity
γ	=	wind turbine yaw angle
σ	=	neural network activation function
Ω	=	parameter space
μ	=	input parameter
$\boldsymbol{\nu}$	=	POD basis
ϵ_{mr}	=	dimensionality reduction error
ϵ_{all}	=	overall prediction error
$\mathbb{I}$	=	indicator function

I. Introduction

Computational fluid dynamics (CFD) is an important tool for investigating complex flow problems. It has been used in many engineering and scientific applications, e.g. aircraft design, weather forecasting, and turbulence research. However, despite the fast development of high-performance computing (HPC) technology, the use of CFD for repetitive

and real-time tasks, such as optimization and real-time control, is still highly challenging due to the requirement of enormous computational resources and long simulation time. For instance, the 3D Reynold-averaged Navier-Stokes (RANS) simulation typically requires hundreds/thousands of CPU hours, not to say the large eddy simulations (LES) and the direct numerical simulations (DNS). Therefore, surrogate modeling, which aims at constructing an efficient yet accurate approximation to the full CFD model, has attracted a lot of attention, such as in aeroelastic computations [1], aerodynamic load evaluations [2], uncertainty quantification of turbulence models [3] and combustion modeling [4].

Reduced basis methods, such as proper orthogonal decomposition (POD) [5–7] and dynamic mode decomposition (DMD) [8], have been used in surrogate modeling. The basic idea of surrogate modeling using reduced basis methods is to decompose the flow dynamics into several reduced basis, which can be interpreted as the coherent structures or modes of the flow. Then the original high-dimensional flow field can be represented by the reduced coefficients of a much lower dimension, with each dimension representing a mode. This type of methods has advantages in physical interpretations and can be used to analyse the energy content and the frequency of the coherent structures. On the other hand, due to the fast development and democratization of machine learning algorithms, surrogate modeling techniques based on machine learning are getting attention in recent years, such as in the aerodynamic simulations with consideration of multiple operating conditions [9] and in the simulations of single-injector combustion process [10].

The surrogate modeling using machine learning techniques can be classified into two categories. One approach is to directly formulate the surrogate modeling of fluid systems as a supervised machine learning problem and employ the state-of-the-art machine learning technique (e.g. deep learning) to train the model [11, 12], then the trained machine learning model can serve directly as the emulator of the original CFD model. This approach can be quite effective if enough training data are available. However, the generation of training data is very expensive as a lot of high-fidelity simulations are required to cover the input parameter space. Another approach is to first reduce the high-dimension flow field into its low-dimension representation, then a supervised machine learning problem is formulated to predict the reduced coefficients of the flow dynamics given the input parameters. The latter approach is the focus of our paper, as it usually requires less training data than the former one.

The combination of POD and supervised machine learning has been investigated recently. In [13], J.S. Hesthaven and S. Ubbiali proposed a surrogate modeling technique called POD-NN, where POD was used to extract the reduced basis and neural network (NN) was used for approximating the map between flow parameters and the POD coefficients. The combination of POD and other machine learning techniques, i.e. Gaussian process regression (GPR), was investigated in [14, 15] and it was tested on a set of numerical examples. The potential to incorporate physical constraints was investigated in [16]. They proposed the use of particular solutions in the POD expansion to enforce certain constraints, e.g. boundary conditions. In addition, a set of supervised machine learning techniques were considered, including NN, multivariate polynomial regression, k-nearest-neighbors, and decision trees. All the work mentioned above employed POD for dimensionality reduction.

has not been investigated yet in the literature. A suitable surrogate modeling method for distributed systems should have the following features: i) the surrogate model can simulate the fluid system with different flow parameters and preferably different layout; ii) the method should be scalable such that the surrogate model can simulate the distributed system of different scales. In this paper, a scalable surrogate modeling method for distributed fluid systems is proposed, where the whole fluid domain is first decomposed into subdomains, then the surrogate modeling for each subdomain is carried out by treating both the boundary information and the distributed flow parameters as the input parameters, and finally the whole flow field is obtained by coupling the flow field of each subdomain altogether with the consideration of the matching condition at the subdomain interface. The information exchange at the subdomain interface needs to be tackled specifically according to the types of the problems. Here, an iterative updating process is introduced for diffusion-dominant problems since each subdomain has an impact on all the other subdomains, while a sequential prediction process is sufficient for convection-dominant problems as the impact of the downstream structures on the upstream flow can be ignored.

Two test cases are used to demonstrate the efficiency, accuracy, and scalability of the proposed surrogate modeling method. The first one is the 1D Poisson's equation which is used to represent the application of the proposed method to diffusion problems. Then a large-scale industrial application, the surrogate modeling of wind farms, is investigated. SOWFA [26], an LES solver developed for the 3-D flow simulation around wind turbine array, is used to generate the high-fidelity data. The application in wind farms aims at capturing the wake interactions between wind turbines, which have a large impact on the plant's overall performance. In the literature, a range of models have been developed to investigate the wake interactions [27], including the high-fidelity LES models and low-fidelity analytical model. In high-fidelity simulations, the turbine rotors were usually represented as actuator disks [28] or actuator lines [29]. The simulations were carried out in [30, 31] using both methods. Although the high-fidelity simulations can predict the turbine wakes accurately, they are too slow for wind farm control design. Low-fidelity wake models are still the main tools for fast wake predictions. Such models include the Jensen model [32, 33], Frandsen model [34], FLORIS model [35], and the model proposed in [36]. In order to retain the reliability of the high-fidelity model while achieving much faster predictions, data-driven surrogate modeling provides an alternative way to model wind farms. The surrogate modeling of a single turbine was studied in [37, 38], where the wake prediction was presented. In [39], the surrogate modeling of two turbine cases was investigated, but the scalability to the wind farm was not considered. In this paper, we apply the proposed surrogate modeling approach for the challenging issue of wind farm wake modeling. The results show that the surrogate model predicts the wind farm wake flow efficiently and accurately, with the root-mean-squared-error being only 2% of the freestream wind speed.

The remaining part of this paper is organized as follows: the machine learning based surrogate modeling method for distributed fluid systems is described in Section II. The proposed method is applied to a diffusion-dominant problem (more specifically, 1D Poisson's equation) and a convection-dominant problem (more specifically, wind farm simulations)

in Section III to demonstrate its scalability, efficiency, and accuracy. Finally the conclusions are drawn in Section IV.

II. Machine Learning Based Surrogate Modeling of Distributed Fluid Systems

In machine learning, dimensionality reduction is usually used to reduce high-dimension training input into its low-dimension representation before feeding it into a regression/classification model. For surrogate modeling of fluid systems, however, the goal is to predict the high-dimension flow field giving a few input parameters. The direct construction of a regression model to predict the high-dimension flow field is prone to overfitting, especially when there are not enough training data available. Therefore, the idea of surrogate modeling for fluid systems is to first reduce the high-dimension flow field into their low-dimension representation, then predict the reduced coefficients from the input parameters, and finally reconstruct the flow field based on the reduced coefficients. One way to achieve this is the use of reduced basis methods, e.g. POD, as is done in [13–16].

We briefly describe the reduced basis method and then illustrate the equivalence of reduced basis method and other machine learning based dimensionality reduction technique in the context of surrogate modeling. Given N samples of input parameters $[\mu_0, \mu_1, \dots, \mu_N]$ and the corresponding CFD flow fields $\mathbf{Z} = [\mathcal{U}_0, \mathcal{U}_1, \dots, \mathcal{U}_N]$, the POD basis $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k, \dots\}$ can be constructed by the singular value decomposition (SVD)

$$\mathbf{Z} = \mathbf{V}\mathbf{\Sigma}\mathbf{W}^T \quad (1)$$

where $\mathbf{v}_k$ is the k^{th} column vector of $\mathbf{V}$. Then after choosing the number of the POD basis as N_r , the flow field can be approximated by

$$\tilde{\mathcal{U}}(\mu_i) = \sum_{k=1}^{N_r} \alpha_k(\mu_i) \mathbf{v}_k \quad (2)$$

where the POD coefficients $\boldsymbol{\alpha}(\mu_i) = [\alpha_1(\mu_i), \alpha_2(\mu_i), \dots, \alpha_{N_r}(\mu_i)]$ are the reduced coefficients of the original flow field $\mathcal{U}_i$. Then a regression model $\mathcal{M}$ can be trained based on the training input $[\mu_0, \mu_1, \dots, \mu_N]$ and the training target $[\boldsymbol{\alpha}(\mu_0), \boldsymbol{\alpha}(\mu_1), \dots, \boldsymbol{\alpha}(\mu_N)]$ such that

$$\mathcal{M}(\mu_i) \approx \boldsymbol{\alpha}(\mu_i) \quad (3)$$

After training, the prediction of the flow field can then be given by

$$\hat{\mathcal{U}}(\mu_{test}) = \sum_{k=1}^{N_r} [\mathcal{M}(\mu_{test})]_k \mathbf{v}_k \quad (4)$$

where μ_{test} is the input parameter of interest. In fact, the reduced basis method here can be viewed as a dimensionality

Algorithm 1 The surrogate modeling method using machine learning

- 1: % **Offline process**
 - 2: Generate N samples of the input parameters: $[\mu_0, \mu_1, \dots, \mu_N]$.
 - 3: Run the high-fidelity CFD solver N times to generate the flow field $\mathbf{Z} = [\mathcal{U}_0, \mathcal{U}_1, \dots, \mathcal{U}_N]$.
 - 4: Use a chosen dimensionality reduction technique to obtain the reduced coefficients $[\mathcal{U}_0^r, \mathcal{U}_1^r, \dots, \mathcal{U}_N^r]$ and the corresponding forward and inverse transforms $\mathbf{g}$ and $\hat{\mathbf{g}}^{-1}$.
 - 5: Train the regression model $\mathcal{M}$ using the training input $[\mu_0, \mu_1, \dots, \mu_N]$ and the training target $[\mathcal{U}_0^r, \mathcal{U}_1^r, \dots, \mathcal{U}_N^r]$.
 - 6: % **Online process**
 - 7: Set the input parameter of interest μ_{test} .
 - 8: Predict the flow field as $\hat{\mathbf{g}}^{-1}(\mathcal{M}(\mu_{test}))$.
-

reduction technique in machine learning, where the forward transform $\mathbf{g}$ and the inverse transform $\hat{\mathbf{g}}^{-1}$ are defined as:

$$[\mathbf{g}(\mathcal{U})]_k = \langle \mathcal{U}, \mathbf{v}_k \rangle, 1 \leq k \leq N_r \quad (5)$$

$$\hat{\mathbf{g}}^{-1}(\alpha) = \sum_{k=1}^{N_r} \alpha_k \mathbf{v}_k \quad (6)$$

where $\langle \rangle$ denotes the inner product. From the above formulation, it is clear that the use of reduced basis methods in surrogate modeling can be replaced by any other dimensionality reduction techniques in machine learning as long as there exist both the forward transform $\mathbf{g}$ which maps the flow field into the reduced coefficients, and the inverse transform $\hat{\mathbf{g}}^{-1}$ which maps the reduced coefficients to the approximation of the flow field. After dimensionality reduction and regression model training, the flow field can then be predicted by

$$\hat{U}(\mu_{test}) = \hat{\mathbf{g}}^{-1}(\mathcal{M}(\mu_{test})) \quad (7)$$

The proposed surrogate modeling procedure is summarized as Algorithm 1. In this work, the fully-connected NN with one hidden layer is chosen as the regression model. POD, ICA, and AE are employed for dimensionality reduction. Hereby these methods are referred as POD-NN, ICA-NN, and AE-NN. The machine learning packages Scikit-learn [40] and Keras[41] are used to facilitate the implementation of the proposed algorithm.

Fig. 2 The illustration of a fully-connected NN with one hidden layer.

Fig. 3 The illustration of an AE with three hidden layers in the NN structure.

A. Regression Model

The regression model used in this work is a fully-connected NN with one hidden layer, as illustrated in Figure 2, and the corresponding input-output relation can be expressed as

$$\begin{aligned} \mathbf{h} &= \sigma(\mathbf{w}_1 \mathbf{x} + \mathbf{b}_1), \\ \mathbf{y} &= \mathbf{w}_2 \mathbf{h} + \mathbf{b}_2 \end{aligned} \quad (8)$$

where $\mathbf{w}_1$, $\mathbf{b}_1$, $\mathbf{w}_2$ and $\mathbf{b}_2$ are the training variables, and σ is the activation function. N_1 , N_2 , and N_3 in Figure 2 represent the input dimension, the hidden-layer neuron number and the output dimension. The NN training process involves the updating of the corresponding weights w_{ij} based on the gradient $\partial J / \partial w_{ij}$, where J is the objective function to be minimized. Automatic differentiation is employed to calculate the gradients. The mean squared error (MSE) between the target and the NN output is chosen as the objective function where a L2 regularization term is further added in order to tackle overfitting. The Adam optimization algorithm [42] is used for NN training in this work.

B. Dimensionality Reduction

Three types of dimensionality reduction techniques (i.e., POD, ICA, and AE) are employed in this paper, to demonstrate the feasibility of using any dimensionality reduction technique with an inverse transform in the proposed surrogate modeling framework. As most of previous studies were based on POD, this work investigates the use of ICA and AE as the alternative dimensionality reduction technique in the context of surrogate modeling. ICA is a popular

technique first developed for signal processing and it can also be used for general purpose dimensionality reduction. The interested readers may refer to [43] for more details. AE is a machine learning technique that makes use of NN to encode a high-dimension input to a low-dimension latent space and then decode it back. It consists of an encoder NN and a decoder NN, and is trained in a self-supervised manner. The AE used in this work is illustrated in Figure 3, where three hidden layers are included in the NN structure. The encoder part can be expressed as

$$\begin{aligned}\mathbf{e} &= \sigma(\mathbf{v}_1\mathbf{x} + \mathbf{a}_1), \\ \mathbf{z} &= \mathbf{v}_2\mathbf{e} + \mathbf{a}_2\end{aligned}\tag{9}$$

where $\mathbf{v}_1$, $\mathbf{a}_1$, $\mathbf{v}_2$ and $\mathbf{a}_2$ are the training variables of the encoder, while the decoder part can be expressed as

$$\begin{aligned}\mathbf{d} &= \sigma(\mathbf{v}_3\mathbf{z} + \mathbf{a}_3), \\ \mathbf{x}^* &= \mathbf{v}_4\mathbf{d} + \mathbf{a}_4\end{aligned}\tag{10}$$

where $\mathbf{v}_3$, $\mathbf{a}_3$, $\mathbf{v}_4$ and $\mathbf{a}_4$ are the training variables of the decoder. M_1 , M_2 , and M_3 in Figure 3 represent the original data dimension, the hidden-layer neuron number and the reduced data dimension. In this work, for simplicity, the hidden-layer neuron number of the encoder and the decoder is assumed to be the same and is set as twice of the reduced dimension. The NN training is carried out using Adam optimization algorithm to minimize the MSE between the NN output and its target value. The target value is the same as the input of the training dataset, thus it is termed as self-supervised training.

C. Extension to Distributed Flow Problems

The surrogate modeling method is extended to tackle distributed flow problems here. A typical example of a distributed fluid system is illustrated in Figure 4, where M cylinders of diameters $\{d_1, d_2, \dots, d_M\}$ are positioned in a rectangular flow domain. The cylinder diameter is the distributed parameter in this example and in fact the distributed parameter can be an array of any other properties of the structures (the surface roughness, thermal conductivity, etc.). The surrogate modeling of this problem is formulated as how to construct a model to predict the whole flow field around the cylinder array given the values of $d_1, d_2, \dots, d_M$. The method proposed in the previous section can be used directly by treating $[d_1, d_2, \dots, d_M]$ as the input parameter μ . However, this approach has two fundamental flaws: i) the training of the surrogate model requires a significant number of CFD evaluations, as the whole flow domain simulation is regarded as a single training sample and the dimension of μ is very high such that a large sample size is required in order to cover the input parameter space. ii) the so-constructed surrogate model is not scalable that it can only be used to simulate the distributed system of the same scale as the training samples.

Fig. 4 A typical example of a distributed fluid system.

Algorithm 2 The surrogate modeling method for distributed flow problems

- 1: **% Offline process**
 - 2: Generate N samples of the input parameters: $[\mu_0, \mu_1, \dots, \mu_N]$, where $\mu_i = [d_i, I_i]$. d_i represents the distributed parameters of a single structure and I_i represents the flow quantities at the subdomain boundary.
 - 3: Run the high-fidelity CFD solver multiple times to obtain the flow fields in a single subdomain $[\mathcal{U}_0, \mathcal{U}_1, \dots, \mathcal{U}_N]$, where $\mathcal{U}_i$ is generated with the distributed parameter d_i and the boundary condition given by I_i .
 - 4: Carry out surrogate modeling using Algorithm 1. Then the flow field in a single subdomain $\tilde{U}_{test}$ can be predicted by the surrogate model given $[d_{test}, I_{test}]$.
 - 5: **% Online process**
 - 6: Set the distributed parameters of all the K structures: $[\tilde{d}_1, \tilde{d}_2, \dots, \tilde{d}_K]$.
 - 7: Initialize the flow quantities at the boundary of each subdomain $[\tilde{I}_1, \tilde{I}_2, \dots, \tilde{I}_K]$.
 - 8: **while** True **do**
 - 9: **for** $j = 1$ to K **do**
 - 10: Predict the flow field $\tilde{U}_j$ in j^{th} subdomain given the input parameter $\tilde{d}_j$ and boundary information $\tilde{I}_j$.
 - 11: **end for**
 - 12: Update $[\tilde{I}_1, \tilde{I}_2, \dots, \tilde{I}_K]$ based on the surrogate model prediction from both sides of the interfaces.
 - 13: **if** the changes of $[\tilde{I}_1, \tilde{I}_2, \dots, \tilde{I}_K]$ are very subtle. **then**
 - 14: The updating process converges. **Break**.
 - 15: **end if**
 - 16: **end while**
 - 17: Combine $[\tilde{U}_1, \tilde{U}_2, \dots, \tilde{U}_K]$ to obtain the prediction of the whole flow field.
-

The scalable surrogate modeling method proposed in our paper can solve these issues. First, the whole flow domain is decomposed into subdomains with each subdomain containing one distributed structure, as illustrated by the dashed rectangular in Figure 4. Then the surrogate modeling is carried out for each subdomain by treating both the distributed parameter of the structure inside the subdomain and the boundary information as the input parameter μ . Finally the flow fields of all the subdomains are combined together with the consideration of the matching condition at the subdomain interface. The proposed approach is inspired naturally by the domain decomposition in high-performance computing in CFD, where the whole domain/mesh is divided into subdomains/blocks and each MPI thread handles only its assigned subdomain with the interface information exchange between MPI threads. The approach can be viewed as employing the surrogate model of individual subdomain to replace the task of each MPI thread, thus greatly reducing the online prediction time. The whole surrogate modeling procedure is summarized as Algorithm 2, where an iterative process is introduced to update the flow quantities at the interface in order to enforce the physical constraints at the interface. The detailed updating rule in Line 12 of Algorithm 2 is problem-dependent. Two numerical examples are given in the rest of the paper with the detailed implementation of the updating rules.

III. Numerical Results

The application of the proposed method on 1D Poisson's equation and on wind farm wake modeling is described in this section. The first test case demonstrates the accuracy, efficiency, and scalability of the proposed surrogate modeling method in diffusion-dominant problems. The second test case further demonstrates the ability of the proposed method in modeling large-scale fluid systems.

A. Application on 1D Poisson's Equation

1. Problem Setup

The 1D steady-state heat transfer with distributed heat sources under consideration is illustrated in Figure 5, where a 1D domain of length l is shown with K uniformly-distributed heat sources of magnitudes $[S_1, S_2, \dots, S_K]$. It can be described by the following equation

$$-\beta \frac{\partial^2 T}{\partial x^2} = q(x) \quad (11)$$

where the heat source term $q = \sum_{i=1}^K S_i \mathbb{I}_{(i-\frac{2}{3})\frac{l}{K} < x < (i-\frac{1}{3})\frac{l}{K}}$ and the boundary condition is given as $T(0) = T_0$ and $T(l) = T_l$. Here $\mathbb{I}$ represents the indicator function. The surrogate modeling of this problem aims at predicting the temperature field efficiently given the boundary conditions (T_0 and T_l) and the distributed parameters $[S_1, S_2, \dots, S_K]$.

In the following, the length of each subdomain l/K is set as 1 and the thermal diffusivity β is set as 1. The FTCS (forward-time central-space) scheme is implemented for numerically solving the equation. The mesh-dependence study

Fig. 5 The 1D steady-state heat transfer problem under consideration.

shows that a uniform mesh of size $1/80$ is sufficient. All calculations are deemed convergent when the root-mean-square error (RMSE) of the temperature profiles between two consecutive time steps is less than 10^{-6} .

2. Results

First the surrogate modeling of a subdomain with a single heat source is carried out. Four hundred samples of the input parameters $[\mu_0, \mu_1, \dots, \mu_{400}]$ in the parameter space $\Omega = [-5.0, 5.0] \times [-5.0, 5.0] \times [-5.0, 5.0]$ are generated using Latin hypercube sampling, where $\mu_i = [S_i, T_{Li}, T_{Ri}]$ with S_i , T_{Li} and T_{Ri} representing the magnitude of the heat source, the temperature at the left boundary and the temperature at the right boundary of the i^{th} sample. Then the Poisson's equation in a domain of length l/K is solved numerically for each sample of the input parameters to generate the temperature profiles. The generated data is then split into two parts, 320 training and validation samples and 80 test samples.

Three surrogate modeling methods, i.e. POD-NN, ICA-NN, AE-NN, are employed to build surrogate models to predict the temperature profile with the magnitude of the heat source, the temperature at the left boundary and the temperature at the right boundary as the model input. The temperatures at both boundaries and heat sources are scaled to zero mean and unit variance separately before feeding into the NN for training. The learning rate is set as 10^{-3} . A grid-search procedure is carried out to determine the optimal hyper-parameters in the dimensionality reduction and regression models, based on 4-fold cross-validation errors. The optimal hyper-parameters, the dimensionality reduction errors, and the prediction errors are given in Table 1 for all the three methods, where N_r represents the dimension of the reduced coefficient, N_h represents the hidden-layer neuron number of the NN regressor, and α_{NN} represents the L2 regularization coefficient of the NN regressor. The optimal hyper-parameter is chosen from the parameter space $\Omega_{N_r \times N_h \times ActFun \times \alpha_{NN}} = \{2, 3, 4, 5, 6, 7\} \times \{4, 6, 8, 10, 12\} \times \{tanh, relu\} \times \{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$. The dimensionality reduction error ϵ_{mr} is defined as the RMSE between $\hat{\mathbf{g}}^{-1}(\mathbf{g}(\mathcal{U}_{test}))$ and $\mathcal{U}_{test}$ and the prediction error ϵ_{all} is defined as the RMSE between $\hat{\mathbf{g}}^{-1}(\mathcal{M}(\mu_{test}))$ and $\mathcal{U}_{test}$, where μ_{test} is the set of test parameters and $\mathcal{U}_{test}$ is the corresponding temperature profiles obtained by the numerical solver. The comparisons of the temperature profiles between the surrogate model predictions and numerical solutions for 4 randomly-selected test cases are shown in Figure 6, including the results given by POD-NN, ICA-NN, and AE-NN. As can be seen, the temperature profiles given by

Fig. 6 The prediction results of a subdomain with a single heat source.

POD-NN match perfectly with the ones given by the numerical solver. Therefore, POD-NN is used in the following for the surrogate modeling of the distributed systems.

Surrogate modeling method	N_r	N_h	Activation function	α_{NN}	ϵ_{mr}	ϵ_{all}
POD-NN	3	10	relu	10^{-6}	2.47×10^{-5}	2.68×10^{-3}
ICA-NN	3	12	relu	10^{-1}	2.47×10^{-5}	5.86×10^{-2}
AE-NN	3	8	relu	10^{-2}	2.41×10^{-3}	9.30×10^{-3}

Table 1 The optimal hyper-parameters of the three surrogate modeling methods for the 1D Poisson case.

For the prediction of the temperature profile of a whole domain with distributed heat sources, the initialization and updating of the temperature at the subdomain boundary need to be specified. Here the temperatures are initialized randomly between T_0 and T_l for the internal subdomain interfaces and the boundary conditions T_0 and T_l are imposed for the most left and the most right subdomains. Then the updating process aims at matching T and $\partial T / \partial x$ at the subdomain interface, as it can be derived that the temperature and the first-order derivative of the temperature need to be

Fig. 7 The prediction results of a domain with 5 uniformly-distributed heat sources.

continuous at the interface.

The results for the case of the domain length $l = 5$ with $K = 5$ heat sources are given here to illustrate the surrogate model's scalability. 100 samples of the distributed parameters and the corresponding numerical solutions are generated in order to assess the accuracy of the surrogate model. The comparisons of the temperature profiles between the surrogate model predictions and numerical solutions for 4 randomly-selected test cases are shown in Figure 7. The RMSE of the surrogate model predictions compared to the high-fidelity numerical solutions averaged over all the test samples is 3.77×10^{-2} . The results clearly demonstrate that the proposed surrogate model predicts the distributed heat transfer problem efficiently and accurately.

B. Surrogate Modeling of Wind Farms

1. Wind Farm Model

The high-fidelity flow field data are needed for the surrogate modeling of wind farms. In this work, the Simulator for Onshore/Offshore Wind Farm Applications (SOWFA) is employed for CFD data generation. SOWFA is a numerical solver developed based on OpenFOAM for the 3-D large eddy simulation of wind flow around wind turbine array in the atmospheric boundary layer, where the turbine rotors are represented by actuator disks or actuator lines. The detailed implementations and the validations of SOWFA can be found in [29, 44].

For wind farm simulations, a precursor simulation of neutral atmospheric boundary layer is first carried out to obtain

Fig. 8 A top view of the simulation domain at turbine hub height.

the initial flow field and inflow boundary conditions, then wind farms are simulated using ALM. A top view of the simulation domain at hub height is given in Figure 8. The size of the simulation domain is $3000m \times 3000m \times 1000m$, with the inflow wind coming from southwest direction. For the mesh generation, a two-level local mesh refinement is used, as is suggested in [45]. The outer mesh dimension is $12m \times 12m \times 12m$, the inner mesh dimension is $3m \times 3m \times 3m$, and the dimension of the mesh in-between is $6m \times 6m \times 6m$. The total number of cells is 1.8×10^7 . In this way, the mesh size around the turbine rotors is 3 meters so that the simulation can capture the detailed turbine wake dynamics. Three NREL 5-MW baseline turbines are positioned in the simulation domain with a 5 rotor diameter spacing in the downstream direction. The rotor diameter of this baseline turbine (denoted as D hereafter) is $126.4m$. For each simulation case, 1500-second simulations are carried out with a time step of $0.02s$. Each case requires around 44 hours

using 256 processors in the local HPC clusters. After LES simulations, the mean velocity field is obtained by averaging the instantaneous flow field from 400s to 1400s. The surrogate modeling in the following part aims at predicting the 2D mean velocity field at turbine hub height efficiently given the inflow conditions and the yaw angles of all the turbines $[\gamma_1, \gamma_2, \dots, \gamma_K]$ for a wind farm consisting of K turbines.

2. Results

First, the surrogate modeling of a subdomain containing a single turbine is carried out. For the high-fidelity data generation, the wind farm simulation is carried out for the case with 3 turbines in a row, as is illustrated in Figure 8. Three inflow wind velocities (i.e., 8m/s, 9m/s, and 10m/s) are considered with a free stream turbulence intensity of 6%. For each wind velocity, 30 samples of turbine yaw angles $[\gamma_0, \gamma_1, \dots, \gamma_{30}]$ in the parameter space $\Omega = [-30.0^\circ, 30.0^\circ] \times [-30.0^\circ, 30.0^\circ] \times [-30.0^\circ, 30.0^\circ]$ are generated using Latin hypercube sampling, where $\gamma_i = [\gamma_i^1, \gamma_i^2, \gamma_i^3]$ with γ_i^1 , γ_i^2 and γ_i^3 representing the yaw angles of the 1st, 2nd, and 3rd turbines of the i^{th} sample. Then SOWFA is used for generating the flow field for each yaw setting. In total, 90 simulations are carried out. The flow field of each subdomain containing one turbine is then extracted as a single data sample. One such subdomain is shown in Figure 8. Therefore, 270 data samples are available for the surrogate modeling of the flow field around one turbine. The data samples are then split into two parts, 216 training and validation samples and 54 test samples.

After completing offline data generation, three surrogate modeling methods, i.e. POD-NN, ICA-NN, AE-NN, are employed to build surrogate models to predict the flow field in a single subdomain with the turbine yaw angle and the wind speeds at 30 uniformly-distributed discrete points along the inflow boundary as the model input. The yaw angles and the inflow velocity profiles are scaled to zero mean and unit variance separately before feeding into the NN for training. The learning rate is set as 10^{-3} . A grid-search procedure is carried out to determine the optimal hyper-parameters in the dimensionality reduction and regression models, based on 4-fold cross-validation errors. The optimal hyper-parameters, the dimensionality reduction errors, and the prediction errors are given in Table 2 for all three methods, where the optimal hyper-parameter is chosen from the parameter space $\Omega_{N_r \times N_h \times ActFun \times \alpha_{NN}} = \{5, 10, 15, 20, 25, 30\} \times \{10, 20, 30, 40, 50\} \times \{tanh, relu\} \times \{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$. The dimensionality reduction error ϵ_{mr} is defined as the RMSE between $\hat{\mathbf{g}}^{-1}(\mathbf{g}(\mathcal{U}_{test}))$ and $\mathcal{U}_{test}$ and the prediction error ϵ_{all} is defined as the RMSE between $\hat{\mathbf{g}}^{-1}(\mathcal{M}(\mu_{test}))$ and $\mathcal{U}_{test}$, where μ_{test} is the set of test inflow conditions and yaw angles and $\mathcal{U}_{test}$ is the corresponding 2D velocity fields obtained by SOWFA.

Surrogate modeling method	N_r	N_h	Activation function	α_{NN}	ϵ_{mr}	ϵ_{all}
POD-NN	15	50	relu	10^{-5}	5.35×10^{-2}	1.67×10^{-1}
ICA-NN	20	50	tanh	10^{-1}	4.45×10^{-2}	1.29×10^{-1}
AE-NN	15	50	relu	10^{-4}	7.74×10^{-2}	1.64×10^{-1}

Table 2 The optimal hyper-parameters of the three surrogate modeling methods for the wind farm case.

Fig. 9 The comparisons between ICA-NN predictions and high-fidelity simulations.

Fig. 10 The comparisons between POD-NN predictions and high-fidelity simulations.

After NN training, the predictions of single turbine wakes are carried out and the results are compared with the SOWFA results for all the 54 test cases. Two typical test cases, the first one with the turbine operating in freestream inflow and the second one with the turbine operating in upstream wake, are shown in Figure 9, 10 and 11, where the flow fields are predicted by ICA-NN, POD-NN and AE-NN respectively. The relative error is defined as the absolute error of the surrogate model prediction divided by the inflow mean velocity. As can be seen, the overall flow fields predicted by all three methods match with the SOWFA results quite well. As the turbine wake evolves from upstream to

Fig. 11 The comparisons between AE-NN predictions and high-fidelity simulations.

downstream, the wake profiles at different streamwise locations, from one rotor diameter ($X = -1D$) in front of the turbine to four rotor diameter ($X = 4D$) behind the turbine, are investigated in order to further examine the prediction performance. As shown in Figure 12(a), a freestream inflow at $X = -1D$ travels downstream and hits the turbine rotor at $X = 0D$. The generated wake then travels from $X = 1D$ to $X = 4D$ in the deflected direction caused by the yaw effects, while slowly recovering toward the freestream conditions and reaching a Gaussian-shape profile at $X = 4D$. At the same time, the wake expands in the spanwise direction as the wake deficit decreases. As for the case shown in Figure 12(b), the inflow at $X = -1D$ is of Gaussian-shape which is generated by the upstream turbines, and the wake development from $X = 0D$ to $X = 4D$ shows similar features as in the freestream case. In summary, the velocity profile predictions match perfectly with the high-fidelity simulation results at all streamwise locations, and all the main flow features such as the wake recovery, the wake deflection and the wake expansion are captured very well by the surrogate model under various turbine yaw and inflow conditions. As ICA-NN is the best among all three methods in terms of prediction error as shown in Table 2, it is used in the following for the surrogate modeling of wind turbine arrays.

The prediction of the flow field around distributed wind turbines follows from Algorithm 2. As the problem here is convection-dominant, the iterative updating process in Algorithm 2 is ignored and the flow quantities at the subdomain boundary are directly imposed according to the upstream surrogate model predictions. Specifically, the turbines' relative positions are first determined according to the wind direction and turbine coordinates. Then the wind fields around the front turbines are predicted by surrogate model with their yaw angles and the freestream inflow profile as the model input. Next, the flow fields around subsequent turbines are predicted similarly, but with the inflow profile extracted from the surrogate model prediction of the corresponding upstream subdomains. In this way, the flow field in the whole wind

Fig. 12 The velocity profiles for the 1-turbine cases, by ICA-NN (dashed) and high-fidelity simulations (solid).

farm can be obtained.

The results of a wind farm with 3 turbines in a row are given here to test the surrogate model's accuracy, efficiency, and scalability. 20 samples of turbine yaw angles are generated and SOWFA is employed to generate the high-fidelity test data in order to assess the accuracy of the surrogate model. The surrogate model predictions and the SOWFA results are only based on the same distributed yaw angles and the same freestream inflow profile. The results of 3 typical test cases are given in Figure 13 where the mean freestream wind speeds are 8 m/s , 9 m/s and 10 m/s respectively. As shown, the prediction errors are quite small in the whole domain for all the test cases. To further examine the surrogate model's performance, the corresponding velocity profiles at various locations from $X = -1D$ to $X = 14D$ are given in Figure 14. As can be seen, the wake deficits generated by the turbines at $X = 0D$, $X = 5D$ and $X = 10D$, the wake development including deflection, recovery and expansion behind each turbine, and the wake interactions between turbines are all successfully predicted. The predicted profiles at all the locations match very well with the corresponding high-fidelity simulation results. Also, the average RMSE of the surrogate model predictions compared to SOWFA results for all 20 test cases is $1.60 \times 10^{-1} \text{ m/s}$, which is just 2% of the freestream wind speed, and the online run time of the surrogate model is negligible (around $9 \times 10^{-4} \text{ s}$ using one core) compared to SOWFA which requires around 44 hours using 256 cores for each case. Therefore, it is concluded that the surrogate model predicts the wind farm flow field efficiently and accurately.

To further illustrate the scalability of the surrogate model, the prediction of a wind farm with 5×5 wind turbines is carried out and the results are given in Figure 15. The inflow velocity is set as 9 m/s and the turbine yaw angles from left to right are set as $[28.4^\circ, 13.3^\circ, -7.63^\circ, -8.50^\circ, -28.1^\circ]$ for the 1st row, $[27.9^\circ, 8.50^\circ, 21.4^\circ, -29.4^\circ, 25.1^\circ]$ for the 2nd row, $[19.5^\circ, 16.5^\circ, -26.5^\circ, -9.15^\circ, -10.8^\circ]$ for the 3rd row, $[-15.7^\circ, 6.54^\circ, -12.1^\circ, -14.6^\circ, -16.9^\circ]$ for the 4th row, and $[16.7^\circ, -2.79^\circ, -27.9^\circ, 10.8^\circ, 26.9^\circ]$ for the 5th row. As can be seen, the wake interactions and the yaw effects are both captured satisfactorily by the surrogate model. Because the so-constructed surrogate model can achieve fast approximation of the

Fig. 13 The comparisons between ICA-NN predictions and high-fidelity simulations for 3-turbine cases.

Fig. 14 The velocity profiles for the 3-turbine cases, by ICA-NN (dashed) and high-fidelity simulations (solid).

original high-fidelity LES model, it can be used directly for the control design of large-scale wind farms.

Fig. 15 The wake prediction of a wind farm with 5×5 wind turbines.

IV. Conclusions

In this paper, a machine learning based surrogate modeling method was proposed, where three dimensionality reduction techniques (i.e. POD, ICA, AE) were investigated for reducing the flow field dimension and neural network was used to predict the reduced coefficients from the input parameters. The surrogate modeling method was specifically designed to tackle distributed fluid systems, by carrying out surrogate modeling for each subdomain and combining the flow field of each subdomain with the consideration of the matching condition at the interface. The applications to a diffusion-dominant problem (more specifically, 1D Poisson's equation) and a convection-dominant problem (more specifically, wind farm simulations) demonstrated the efficiency, accuracy, and scalability of the proposed surrogate modeling method. In particular, the surrogate model of wind farm wakes predicted the wind farm velocity field very accurately with an average root-mean-square error (compared to high-fidelity results) being 2% of the freestream wind speed, while the online prediction time is negligible (around $9 \times 10^{-4} s$ using one core) compared to high-fidelity simulation which requires around 44 hours using 256 cores for each case. This demonstrated the ability of the proposed method in modeling large-scale distributed fluid systems.

Future research may involve the application of the proposed surrogate modeling to other flow problems, such as the

flow field around heater arrays in heat exchangers and the distributed roughness elements in boundary layer flow. The optimal design using the constructed surrogate model is also of great interest. For example, it can be used for wind farm layout optimization or as an internal model for wind farm yaw optimization. Another research direction is the surrogate modeling of unsteady distributed systems and the optimal control design based on such surrogate models.

Acknowledgments

This work has received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 765579. The authors also acknowledge the Scientific Computing Research Technology Platform (SCRTP) at University of Warwick for providing High Performance Computing resources.

References

- [1] Amsallem, D., Cortial, J., and Farhat, C., “Towards Real-Time Computational-Fluid-Dynamics-Based Aeroelastic Computations Using a Database of Reduced-Order Information,” *AIAA Journal*, Vol. 48, No. 9, 2010, pp. 2029–2037. <https://doi.org/10.2514/1.j050233>.
- [2] Fossati, M., “Evaluation of Aerodynamic Loads via Reduced-Order Methodology,” *AIAA Journal*, Vol. 53, No. 8, 2015, pp. 2389–2405. <https://doi.org/10.2514/1.j053755>.
- [3] Zhang, J., and Fu, S., “An efficient approach for quantifying parameter uncertainty in the SST turbulence model,” *Computers & Fluids*, Vol. 181, 2019, pp. 173–187. <https://doi.org/10.1016/j.compfluid.2019.01.017>.
- [4] Wang, Q., Hesthaven, J. S., and Ray, D., “Non-intrusive reduced order modeling of unsteady flows using artificial neural networks with application to a combustion problem,” *Journal of Computational Physics*, Vol. 384, 2019, pp. 289–307. <https://doi.org/10.1016/j.jcp.2019.01.031>.
- [5] Berkooz, G., Holmes, P., and Lumley, J. L., “The Proper Orthogonal Decomposition in the Analysis of Turbulent Flows,” *Annual Review of Fluid Mechanics*, Vol. 25, No. 1, 1993, pp. 539–575. <https://doi.org/10.1146/annurev.fl.25.010193.002543>.
- [6] Chatterjee, A., “An introduction to the proper orthogonal decomposition,” *Current science*, Vol. 78, No. 7, 2000, pp. 808–817. URL <http://www.jstor.org/stable/24103957>.
- [7] LIANG, Y., LEE, H., LIM, S., LIN, W., LEE, K., and WU, C., “PROPER ORTHOGONAL DECOMPOSITION AND ITS APPLICATIONS—PART I: THEORY,” *Journal of Sound and Vibration*, Vol. 252, No. 3, 2002, pp. 527–544. <https://doi.org/10.1006/jsvi.2001.4041>.
- [8] SCHMID, P. J., “Dynamic mode decomposition of numerical and experimental data,” *Journal of Fluid Mechanics*, Vol. 656, 2010, pp. 5–28. <https://doi.org/10.1017/s0022112010001217>.
- [9] Dupuis, R., Jouhaud, J.-C., and Sagaut, P., “Surrogate Modeling of Aerodynamic Simulations for Multiple Operating Conditions Using Machine Learning,” *AIAA Journal*, Vol. 56, No. 9, 2018, pp. 3622–3635. <https://doi.org/10.2514/1.j056405>.

- [10] Swischuk, R., Kramer, B., Huang, C., and Willcox, K., "Learning Physics-Based Reduced-Order Models for a Single-Injector Combustion Process," *AIAA Journal*, Vol. 58, No. 6, 2020, pp. 2658–2672. <https://doi.org/10.2514/1.j058943>.
- [11] Guo, X., Li, W., and Iorio, F., "Convolutional Neural Networks for Steady Flow Approximation," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2016. <https://doi.org/10.1145/2939672.2939738>.
- [12] Jin, X., Cheng, P., Chen, W.-L., and Li, H., "Prediction model of velocity field around circular cylinder over various Reynolds numbers by fusion convolutional neural networks based on pressure on the cylinder," *Physics of Fluids*, Vol. 30, No. 4, 2018, p. 047105. <https://doi.org/10.1063/1.5024595>.
- [13] Hesthaven, J., and Ubbiali, S., "Non-intrusive reduced order modeling of nonlinear problems using neural networks," *Journal of Computational Physics*, Vol. 363, 2018, pp. 55–78. <https://doi.org/10.1016/j.jcp.2018.02.037>.
- [14] Guo, M., and Hesthaven, J. S., "Reduced order modeling for nonlinear structural analysis using Gaussian process regression," *Computer Methods in Applied Mechanics and Engineering*, Vol. 341, 2018, pp. 807–826. <https://doi.org/10.1016/j.cma.2018.07.017>.
- [15] Guo, M., and Hesthaven, J. S., "Data-driven reduced order modeling for time-dependent problems," *Computer Methods in Applied Mechanics and Engineering*, Vol. 345, 2019, pp. 75–99. <https://doi.org/10.1016/j.cma.2018.10.029>.
- [16] Swischuk, R., Mainini, L., Peherstorfer, B., and Willcox, K., "Projection-based model reduction: Formulations for physics-based machine learning," *Computers & Fluids*, Vol. 179, 2019, pp. 704–717. <https://doi.org/10.1016/j.compfluid.2018.07.021>.
- [17] Hyvärinen, A., Karhunen, J., and Oja, E., *Independent Component Analysis*, John Wiley & Sons, Inc., 2001. <https://doi.org/10.1002/0471221317>.
- [18] Hinton, G. E., "Reducing the Dimensionality of Data with Neural Networks," *Science*, Vol. 313, No. 5786, 2006, pp. 504–507. <https://doi.org/10.1126/science.1127647>.
- [19] Tou, S., Tso, C., and Zhang, X., "3-D numerical analysis of natural convective liquid cooling of a 3×3 heater array in rectangular enclosures," *International Journal of Heat and Mass Transfer*, Vol. 42, No. 17, 1999, pp. 3231–3244. [https://doi.org/10.1016/s0017-9310\(98\)00379-2](https://doi.org/10.1016/s0017-9310(98)00379-2).
- [20] Mon, M. S., and Gross, U., "Numerical study of fin-spacing effects in annular-finned tube heat exchangers," *International Journal of Heat and Mass Transfer*, Vol. 47, No. 8-9, 2004, pp. 1953–1964. <https://doi.org/10.1016/j.ijheatmasstransfer.2003.09.034>.
- [21] Rizzetta, D. P., and Visbal, M. R., "Direct Numerical Simulations of Flow Past an Array of Distributed Roughness Elements," *AIAA Journal*, Vol. 45, No. 8, 2007, pp. 1967–1976. <https://doi.org/10.2514/1.25916>.
- [22] Liu, J., Srebric, J., and Yu, N., "Numerical simulation of convective heat transfer coefficients at the external surfaces of building arrays immersed in a turbulent boundary layer," *International Journal of Heat and Mass Transfer*, Vol. 61, 2013, pp. 209–225. <https://doi.org/10.1016/j.ijheatmasstransfer.2013.02.005>.

- [23] Calaf, M., Meneveau, C., and Meyers, J., “Large eddy simulation study of fully developed wind-turbine array boundary layers,” *Physics of Fluids*, Vol. 22, No. 1, 2010, p. 015110. <https://doi.org/10.1063/1.3291077>.
- [24] Wu, Y.-T., and Porté-Agel, F., “Modeling turbine wakes and power losses within a wind farm using LES: An application to the Horns Rev offshore wind farm,” *Renewable Energy*, Vol. 75, 2015, pp. 945–955. <https://doi.org/10.1016/j.renene.2014.06.019>.
- [25] Nilsson, K., Ivanell, S., Hansen, K. S., Mikkelsen, R., Sørensen, J. N., Breton, S.-P., and Henningson, D., “Large-eddy simulations of the Lillgrund wind farm,” *Wind Energy*, Vol. 18, No. 3, 2014, pp. 449–467. <https://doi.org/10.1002/we.1707>.
- [26] “NWTC Information Portal (SOWFA),” <https://nwtc.nrel.gov/SOWFA>, 2014.
- [27] Boersma, S., Doekemeijer, B., Gebraad, P., Fleming, P., Annoni, J., Scholbrock, A., Frederik, J., and van Wingerden, J.-W., “A tutorial on control-oriented modeling and control of wind farms,” *2017 American Control Conference (ACC)*, IEEE, 2017. <https://doi.org/10.23919/acc.2017.7962923>.
- [28] Meyers, J., and Meneveau, C., “Large Eddy Simulations of Large Wind-Turbine Arrays in the Atmospheric Boundary Layer,” *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, American Institute of Aeronautics and Astronautics, 2010. <https://doi.org/10.2514/6.2010-827>.
- [29] Churchfield, M. J., Lee, S., Michalakes, J., and Moriarty, P. J., “A numerical study of the effects of atmospheric and wake turbulence on wind turbine dynamics,” *Journal of Turbulence*, Vol. 13, 2012, p. N14. <https://doi.org/10.1080/14685248.2012.668191>.
- [30] Witha, B., Steinfeld, G., and Heinemann, D., “High-Resolution Offshore Wake Simulations with the LES Model PALM,” *Research Topics in Wind Energy*, Springer Berlin Heidelberg, 2014, pp. 175–181. https://doi.org/10.1007/978-3-642-54696-9_26.
- [31] Martínez-Tossas, L. A., Churchfield, M. J., and Leonardi, S., “Large eddy simulations of the flow past wind turbines: actuator line and disk modeling,” *Wind Energy*, Vol. 18, No. 6, 2014, pp. 1047–1060. <https://doi.org/10.1002/we.1747>.
- [32] Jensen, N. O., *A note on wind generator interaction*, 1983.
- [33] Katic, I., Højstrup, J., and Jensen, N. O., “A simple model for cluster efficiency,” *European wind energy association conference and exhibition*, A. Raguzzi, 1987.
- [34] Frandsen, S., Barthelmie, R., Pryor, S., Rathmann, O., Larsen, S., Højstrup, J., and Thøgersen, M., “Analytical modelling of wind speed deficit in large offshore wind farms,” *Wind Energy*, Vol. 9, No. 1-2, 2006, pp. 39–53. <https://doi.org/10.1002/we.189>.
- [35] Gebraad, P. M. O., Teeuwisse, F. W., van Wingerden, J. W., Fleming, P. A., Ruben, S. D., Marden, J. R., and Pao, L. Y., “Wind plant power optimization through yaw control using a parametric model for wake effects-a CFD simulation study,” *Wind Energy*, Vol. 19, No. 1, 2014, pp. 95–114. <https://doi.org/10.1002/we.1822>.
- [36] Niayifar, A., and Porté-Agel, F., “Analytical Modeling of Wind Farms: A New Approach for Power Prediction,” *Energies*, Vol. 9, No. 9, 2016, p. 741. <https://doi.org/10.3390/en9090741>.

- [37] Iungo, G. V., Santoni-Ortiz, C., Abkar, M., Porté-Agel, F., Rotea, M. A., and Leonardi, S., “Data-driven Reduced Order Model for prediction of wind turbine wakes,” *Journal of Physics: Conference Series*, Vol. 625, 2015, p. 012009. <https://doi.org/10.1088/1742-6596/625/1/012009>.
- [38] Hamilton, N., Viggiano, B., Calaf, M., Tutkun, M., and Cal, R. B., “A generalized framework for reduced-order modeling of a wind turbine wake,” *Wind Energy*, Vol. 21, No. 6, 2018, pp. 373–390. <https://doi.org/10.1002/we.2167>.
- [39] Annoni, J., Gebraad, P., and Seiler, P., “Wind farm flow modeling using an input-output reduced-order model,” *2016 American Control Conference (ACC)*, IEEE, 2016. <https://doi.org/10.1109/acc.2016.7524964>.
- [40] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E., “Scikit-Learn: Machine Learning in Python,” *Journal of machine learning research*, Vol. 12, No. Nov, 2011, p. 2825–2830. URL <https://dl.acm.org/doi/10.5555/1953048.2078195>.
- [41] Chollet, F., et al., “Keras,” <https://github.com/fchollet/keras>, 2015.
- [42] Kingma, D. P., and Ba, J., “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [43] Hyvärinen, A., and Oja, E., “Independent component analysis: algorithms and applications,” *Neural Networks*, Vol. 13, No. 4-5, 2000, pp. 411–430. [https://doi.org/10.1016/s0893-6080\(00\)00026-5](https://doi.org/10.1016/s0893-6080(00)00026-5).
- [44] Churchfield, M., Lee, S., Moriarty, P., Martinez, L., Leonardi, S., Vijayakumar, G., and Brasseur, J., “A Large-Eddy Simulations of Wind-Plant Aerodynamics,” *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, American Institute of Aeronautics and Astronautics, 2012. <https://doi.org/10.2514/6.2012-537>.
- [45] Fleming, P., Gebraad, P. M., Lee, S., van Wingerden, J.-W., Johnson, K., Churchfield, M., Michalakes, J., Spalart, P., and Moriarty, P., “Simulation comparison of wake mitigation control strategies for a two-turbine case,” *Wind Energy*, Vol. 18, No. 12, 2014, pp. 2135–2143. <https://doi.org/10.1002/we.1810>.