Computational Mechanics (2023) 71:481–491
https://doi.org/10.1007/s00466-022-02251-1

ORIGINAL PAPER

Physics-informed machine learning for surrogate modeling of wind pressure and optimization of pressure sensor placement

Qiming Zhu[1] · Ze Zhao[1] · Jinhui Yan[1]

Received: 21 May 2022 / Accepted: 6 November 2022 / Published online: 21 November 2022

Abstract
This paper presents a predictive computational framework for surrogate modeling of pressure field and optimization of pressure sensor placement for wind engineering applications. Firstly, a machine learning-derived surrogate model, trained by high-fidelity simulation data using finite element-based CFD and informed by a turbulence model, is developed to construct the full-field pressure from scattered sensor measurements in near real-time. Then, the surrogate pressure model is embedded in another neural network (NN) for optimizing pressure sensor placement. The goal of the NN-based optimizer is to learn the best layout of a fixed number of pressure sensors over the structural surface to deliver the most accurate full-field pressure prediction for various inflow wind conditions. We deploy the model to a representative low-rise building subjected to different wind conditions. The performance of the proposed framework is assessed by comparing the predicted results with finite element-based CFD simulation results. The framework shows excellent accuracy and efficiency, which could be potentially integrated with structural health monitoring to enable digital twins of civil structures.

Keywords CFD · Machine learning · Finite element for fluid mechanics

1 Introduction

Civil structures in the U.S. coastal areas, home to more than 127 million people, have always been a critical concern because of frequent extreme wind events. Predicting the pressure field on structures is vital for responding to and mitigating threats of extreme wind events to civil structures. In recent years, numerical simulations based on computational fluid dynamics (CFD) are becoming prevalent in structural/wind engineering research communities because of their systematic, repeatable, controllable boundary conditions and their ability to estimate wind-induced pressure loads. Two popular CFD approaches are Reynolds-averaged Navier–Stokes (RANS) and large-eddy simulations (LES). Despite the distinct turbulence models used, the core of RANS and LES is based on numerical discretizations of the respective governing equations (e.g., time-averaged Navier–Stokes equations with closure model in RANS and filtered Navier–Stokes equations in LES) or their weak forms. These two approaches will continue to play essential roles in civil engineering practice. However, they have two critical limitations for wind-induced pressure predictions. Firstly, these two methods are computationally expensive and sometimes take longer time than wind tunnel tests to simulate the same physical time length. With the surge of interest in developing and adopting digital twins, employing these models in digital twins that require real-time prediction is challenging for large-scale civil structures due to the high computational cost. Secondly, both methods don't fully harness the value of labeled and high-fidelity simulation data, which are primarily used for verification and validation purposes. Embedding the labeled data into the predictive models generated by these methods to extrapolate to new wind scenarios is extremely challenging.

Recent years have witnessed the boom of machine learning (ML) and data-driven models in engineering design and analysis, such as material design, cardiovascular modeling, heat transfer, water resources, and advanced manufacturing [1–17]. Among many ML models, scientific machine learning (SciML) has shown significant potential to accelerate scientific predictions by harnessing data from experiments and high-fidelity physics-based simulations. One

✉ Jinhui Yan
yjh@illinois.edu

1 Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, Champaign, IL, USA

widely used approach of SciML is to train a conventional deep learning (DL) model such as Gaussian process regression (GPR), or deep neural network (NN) informed physical principles, e.g., conservation laws and constitutive relationships. These conservation laws are usually represented by partial differential equations (PDEs). One significant advantage of these physics-informed machine learning models is that they don't require big-data owing to the highly condensed knowledge embedded in the physical principles. Existing research has demonstrated SciML's capability in various engineering areas. In particular, SciML has been proved to be a powerful tool for developing surrogate models and reduced-order models (ROMs) with well-balanced accuracy and efficiency using sparse training data [12] (even without training data [18]).

In light of these advances, a predictive SciML-based computational framework for wind-induced pressure loads is proposed in this paper. The SciML framework has two main functionalities. The first is a surrogate model for wind pressure derived via a physics-informed neural network (PINN). The surrogate model can quickly recover the full-field pressure profile from scattered measurements, maintaining high accuracy while causing less computational cost compared with LES and RANS models. Nowadays, many smart homes are equipped with installed sensors. The surrogate model could be potentially integrated with the monitoring system such that homeowners can know the wind loads on their roofs in real-time. The second is an ML-based optimizer using the PINN-based surrogate model as a fast evaluation tool to learn the best placement for a given number of pressure sensors to achieve the best predictive accuracy over a wide range of wind conditions. The sensor placement optimizer allows users to maximize the value of a limited budget with a given number of pressure sensors. To demonstrate the effectiveness of the proposed framework, we utilize a RANS model using a stabilized finite element method (FEM) to generate synthetic training and validation data. A classical flat roof is selected for investigation in this paper.

The remaining paper is organized as follows. Section 2 presents the finite element-based RANS model. Section 3 presents the details of the problem considered in this paper and the training/testing data generation. Section 4 describes the mathematical details of the pressure surrogate model and the optimization algorithm for the pressure sensor placement. The ML architecture and training method used in the framework are also described in this section. Section 5 analyzes the effects of sensor location and number. The conclusions are drawn in Sect. 6.

2 High-fidelity model

Let Ω denote the problem domain. The wind velocity $\boldsymbol{u}$ and wind pressure p in Ω are governed by the following Navier–Stokes equations of incompressible flows

$$\rho\left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\nabla\boldsymbol{u} - \boldsymbol{g}\right) - \nabla\cdot\boldsymbol{\sigma} = \boldsymbol{0} \qquad \text{in } \Omega \tag{1}$$

$$\nabla\cdot\boldsymbol{u} = 0 \qquad \text{in } \Omega \tag{2}$$

where ρ is the fluid density, $\boldsymbol{g}$ is the gravitational acceleration, $\boldsymbol{\sigma} = (-p\boldsymbol{I} + 2\mu\nabla^s\boldsymbol{u})$ is the Cauchy stress tensor, in which μ is the dynamic viscosity, ∇^s is the symmetric part of gradient operator, and $\boldsymbol{I}$ is a 3×3 identity tensor.

Equations 1 and 2 are subjected to following Dirichlet and Neumann boundary conditions

$$\mathcal{L}_{BC}(\boldsymbol{u}, p) = \begin{cases} \boldsymbol{u} - \boldsymbol{u}_{bc} &= \boldsymbol{0} \qquad \text{on } \Gamma_D \\ \boldsymbol{\sigma}\cdot\boldsymbol{n} - \boldsymbol{h} &= \boldsymbol{0} \qquad \text{on } \Gamma_N \end{cases} \tag{3}$$

where $\boldsymbol{u}_{bc}$ is the prescribed velocity on Dirichlet boundary Γ_D (e.g., inlet), $\boldsymbol{h}$ is the prescribed fluid traction on the Neumann boundary Γ_N, and $\boldsymbol{n}$ is the unit normal vector on the boundary.

This paper focuses on predicting mean wind pressure profiles on structural surfaces. Given the large Reynolds (Re) number considered, a classical time-independent Reynolds-averaged Navier–Stokes (RANS) approach using a mixing length algebraic turbulence closure model is adopted. The RANS equations are given as

$$\mathcal{L}_M(\bar{\boldsymbol{u}}, p) = \rho\left[\bar{\boldsymbol{u}}\cdot\nabla\bar{\boldsymbol{u}} - \boldsymbol{g}\right] - \nabla\cdot\boldsymbol{\sigma_t} \quad = \boldsymbol{0} \qquad \text{in } \Omega \tag{4}$$

$$\mathcal{L}_C(\bar{\boldsymbol{u}}) = \nabla\cdot\bar{\boldsymbol{u}} = 0 \qquad \text{in } \Omega \tag{5}$$

where $\bar{\boldsymbol{u}}$ and $\bar{p}$ are the Reynolds-averaged velocity and pressure fields. $\boldsymbol{\sigma_t} = -\bar{p}\boldsymbol{I} + 2\mu_t\nabla^s\bar{\boldsymbol{u}}$ is the turbulent Reynolds stress tensor, where $\mu_t = \mu + \rho L_{mix}^2\sqrt{2\nabla^s\bar{\boldsymbol{u}} : \nabla^s\bar{\boldsymbol{u}}}$ is the turbulent viscosity, in which $L_{mix} = \kappa y$ is a mixing length with κ being the Von Karman constant and y being the minimum distance to the wall.

The above RANS model is solved by a stabilized finite element method based on Streamline upwind Petrov–Galerkin (SUPG) and Pressure-Stabilizing Petrov–Galerkin (PSPG) formulations [19–21]. Let $\boldsymbol{V}_u$ and $\boldsymbol{V}_p$ denote discrete velocity, and pressure trial function spaces, and $\boldsymbol{W}_u$ and $\boldsymbol{W}_p$ denote the corresponding test function spaces. The weak form of the RANS model is stated as follows. Find $\bar{\boldsymbol{u}} \in \boldsymbol{V}_u$ and $\bar{p} \in \boldsymbol{V}_p$ such that for $\forall\ \bar{\boldsymbol{w}} \in \boldsymbol{W}_u$ and $\bar{q} \in \boldsymbol{W}_p$

$$B\left(\{\bar{\boldsymbol{w}}, \bar{q}\}, \{\bar{\boldsymbol{u}}, \bar{p}\}\right) - F\left(\{\bar{\boldsymbol{w}}, \bar{q}\}\right) = 0 \tag{6}$$

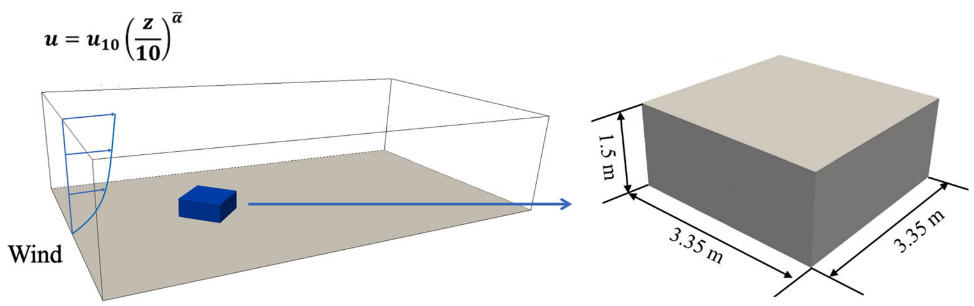


Fig. 1 Problem setup. Left: Wind flow pass a flat roof. Right: Dimensions of the roof

where $B\left(\{\bar{\boldsymbol{w}}, \bar{q}\}, \{\bar{\boldsymbol{u}}, \bar{p}\}\right)$ and $F\left(\{\bar{\boldsymbol{w}}, \bar{q}\}\right)$ are given as

$$\begin{aligned} B\left(\{\bar{\boldsymbol{w}}, \bar{q}\}, \{\bar{\boldsymbol{u}}, \bar{p}\}\right) = & \int_{\Omega} \bar{\boldsymbol{w}} \cdot \rho \left(\bar{\boldsymbol{u}} \cdot \nabla \bar{\boldsymbol{u}}\right) \, d\Omega \\ & + \int_{\Omega} \nabla^{s} \bar{\boldsymbol{w}} : \boldsymbol{\sigma}_t \left(\bar{\boldsymbol{u}}, \bar{p}\right) \, d\Omega + \int_{\Omega} \bar{q} \, \nabla \cdot \bar{\boldsymbol{u}} \, d\Omega \\ & + \int_{\Omega} \left(\bar{\boldsymbol{u}} \cdot \nabla \bar{\boldsymbol{w}} + \frac{\nabla \bar{q}}{\rho}\right) \cdot \left[\tau_M \boldsymbol{\mathcal{L}}_M (\bar{\boldsymbol{u}}, \bar{p})\right] \, d\Omega \\ & + \int_{\Omega} \rho \tau_C \nabla \cdot \bar{\boldsymbol{w}} \mathcal{L}_C(\bar{\boldsymbol{u}}) \, d\Omega \end{aligned} \tag{7}$$

$$F\left(\{\bar{\boldsymbol{w}}, \bar{q}\}\right) = \int_{\Omega} \bar{\boldsymbol{w}} \cdot \rho \, \boldsymbol{g} \, d\Omega + \int_{\Gamma_N} \bar{\boldsymbol{w}} \cdot \boldsymbol{h} \, dA \tag{8}$$

where τ_M and τ_C are the SUPG and PSPG-based stabilization parameters. Their definitions and related discussions can be found in [19–24]. Although the stabilized FEM formulation is deployed to a steady RANS model in this paper, it is noteworthy to mention that the formulation and its more advanced versions, such as Arbitrary Lagrangian-Eulerian technique (ALE-VMS) [25–32] and Space-Time (ST-VMS) technique [33–37], have successfully been employed as large eddy simulation (LES) models in simulating of a wide range of challenging fluid dynamics and fluid-structure interaction problems. These methods show significant advantages when being deployed to flow problems with moving interfaces and boundaries. Several recent validations and applications include environmental flows [38–41], wind energy [28,42–60], tidal energy [58,61–65], cavitation flows [66,67], supersonic flows [68], bio-mechanics [69–74], gas turbine [75–77], and transportation engineering [23,78–82].

The velocity and pressure fields in Eq. 6 are solved in a fully coupled fashion. The nonlinear equations are linearized by the Newton's method. The resulting linear systems are solved by a generalized minimal residual method with a block preconditioning technique [83]. The formulation is implemented in parallel for high-performance computing environments using the Message Passing Interface (MPI).

3 Problem statement and data generation

The major objective of the developed ML framework is to enable fast full-field pressure prediction and maximize the value with a limited number of pressure sensors by optimizing their locations. To demonstrate the effectiveness of the ML framework, we apply it to wind flows past a low-rise building roof, which represents the most vulnerable structures to extreme wind events. Figure 1 shows the problem considered in this paper, in which a flat roof of a low-rise building with dimension of 3.35 m × 3.35 m × 1.5 m is simulated with different wind flows in the direction of 45 degree. The reason for choosing 45 degree for study is that it is the most critical angle for low-rise buildings. Past studies show that this wind direction can generate high uplifts on flat rectangular roofs due to conical vortices [84–87].

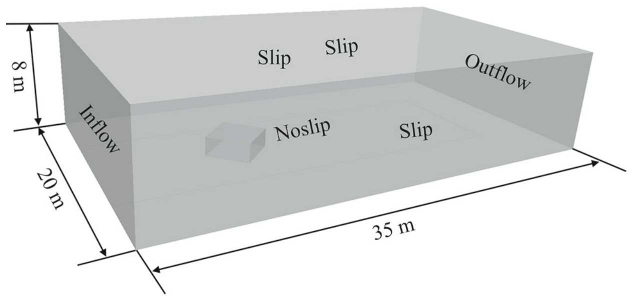


Fig. 2 Setup and boundary conditions

Following the American Society of Civil Engineers (ASCE) 7–10 code, the roof is subjected to the following wind flow profile obeying a power law, namely,

$$\overline{\boldsymbol{U}}_z = \boldsymbol{u}_{10}\left(\frac{z}{10}\right)^{\overline{\alpha}} \tag{9}$$

where $\overline{\boldsymbol{U}}_z$ is the hourly mean wind speed, z is the height, $\boldsymbol{u}_{10}$ is the mean hourly wind speed at $z = 10\ m$ height, $\overline{\alpha}$ is a constant related with the type of exposure influenced by the characteristics of the ground roughness and surface irregularities in the vicinity of the building.

We utilize the FEM-based RANS model presented in Sect. 2 to generate simulation data to train and test the ML framework. Figure 2 shows the simulation setup and boundary conditions (BCs). The computational domain is a box with dimensions of 35 m × 20 m × 8 m. At the inlet, a strong BC using the wind profile based on Eq. 9 is prescribed. At the outlet, traction-free outflow BC is used. Slip and no-penetration BC is utilized in the span-wise direction and on the top surface. At last, no-slip BC is utilized on the bottom and building surfaces. The RANS simulations make use of thin triangular prism elements to capture the boundary layers around the roof and linear tetrahedron elements elsewhere. For bluff structures, flow separation and vortex happen near the structure and in the wake. To better capture the physics without increasing cost, we do local refinements with three different levels of mesh size in the computational domain. The finest mesh size near the roof surface is 0.02 m. The medium mesh size near a refined interior box is 0.2 m. At last, the coarsest mesh size in the outer box is 2.0 m. The final CFD mesh consists of 3,819,670 nodes, 16,197,989 tetrahedron elements, and 2,000,488 prism elements. Figure 3 depicts several snapshots of the mesh. In the RANS simulations, the density of the air is set to $1.225\ \mathrm{kg\,m^{-3}}$, and dynamic viscosity is set to $1.8 \times 10^{-5}\ \mathrm{kg\,m^{-1}s^{-1}}$.

We utilize the above CFD model to simulate 5×5 different combinations of $\boldsymbol{u}_{10}$ and $\overline{\alpha}$ for the training and testing of the ML framework. These cases cover a wide range of wind conditions in the US. Among these cases, 11 are used for training the surrogate model, 11 cases are used for training the optimizer of the pressure sensor placement, and 3 cases are used for testing the ML model's predictive accuracy. The specification of these cases can be found in Fig. 4.

4 Physics-informed ML framework for pressure prediction and sensor placement optimization

As shown in Fig. 5, the ML framework consists of a PINN-derived surrogate predictive model for full-field pressure and a neural network based optimizer for pressure sensor placement. The objective of the ML framework is two-fold. The first is to quickly and accurately recover the full-field pressure profile by only using a small number of scattered pressure measurements for a given inflow wind profile. The second is to find the optimal placement of a fixed number of pressure sensors, which delivers the most accurate full-field pressure predictions for a set of inflow wind conditions. We should mention that the novelty of our framework is not in the network structure.. Instead, we aim to use a physics-informed neural network to design a two-step model in order to build a pressure surrogate model and a pressure sensor location optimizer, which could be used in digital twins for civil structures.

4.1 PINN-driven surrogate model

A physics-informed neural network (PINN) has been utilized to develop the surrogate pressure model [88]. PINN has shown strong capabilities in solving forward and inverse problems involving nonlinear partial differential equations and deriving surrogate and reduced-order models by mixing labeled data and physical principles. In general, neural network is a computing architecture that is vaguely inspired by the biological neural networks constituting animal brains [89]. Typical neural network architectures include fully connected neural network (FCNN) [90], convolutional neural network (CNN) [91], and recurrent neural network (RNN) [92]. A neural network with more than one hidden layer is conventionally called a deep neural network, whose function approximation capability increases with the number of hidden layers and neurons [93]. Activation functions are often used to introduce the non-linearity to the NN [94]. Widely used activation functions in deep learning are tanh function, rectified linear unit (Relu) function, and sigmoid function [95]. In this paper, the PINN employs a fully connected deep neural network [90], where the neurons of adjacent layers are fully connected, and a swish activation function [96]. 8 hidden layers and 150 neurons per hidden layers are used for the pressure surrogate model and the sensor placement optimization model.

The PINN-based surrogate model, which is trained by high-fidelity data and informed by the turbulence model, constructs a full-field pressure field from scattered pressure measurements, namely,

$$[\boldsymbol{x}, \boldsymbol{X}_s, \tilde{P}_s] \xrightarrow{\boldsymbol{W}^*, \boldsymbol{b}^*} [\boldsymbol{u}_{NN}(\boldsymbol{x}), p_{NN}(\boldsymbol{x})] \tag{10}$$

where $\boldsymbol{x}$ is the spatial coordinates of interest, covering the entire roof surface, $\boldsymbol{X}_s = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_{N_s}\}$ are the locations of the pressure sensors, with N_s denoting the number of pressure sensors, $\tilde{P}_s = \{\tilde{p}_1, \tilde{p}_2, ..., \tilde{p}_{N_s}\}$ are the measured pressure at these locations. The output $\boldsymbol{u}_{NN}(\boldsymbol{x})$ and $p_{NN}(\boldsymbol{x})$ are the averaged velocity and full-field pressure predicted by the surrogate model. One should note that the surrogate model also outputs velocity, despite that the primary quantity of interest (QoI) is pressure, and only pressure data is used in training for the wind engineering problem considered in this paper. $\boldsymbol{W}^*$ and $\boldsymbol{b}^*$ are the NN's weights and biases, which need to be learned by solving an optimizing problem, defined as

$$W^*, b^* = \underset{W,b}{\operatorname{argmin}}\ \mathcal{R}_s \tag{11}$$

where $\mathcal{R}_s$ is the objective function, which consists of the following three components

$$\mathcal{R}_s = \lambda_{data}\mathcal{R}_{data} + \lambda_{pde}\mathcal{R}_{pde} + +\lambda_{bc}\mathcal{R}_{bc} \tag{12}$$

where $\mathcal{R}_{data}$, $\mathcal{R}_{pde}$, and $\mathcal{R}_{bc}$ represent the data-driven component, physics-informed components in the interior domain and boundary conditions, respectively. λ_{data}, λ_{pde}, and λ_{bc}

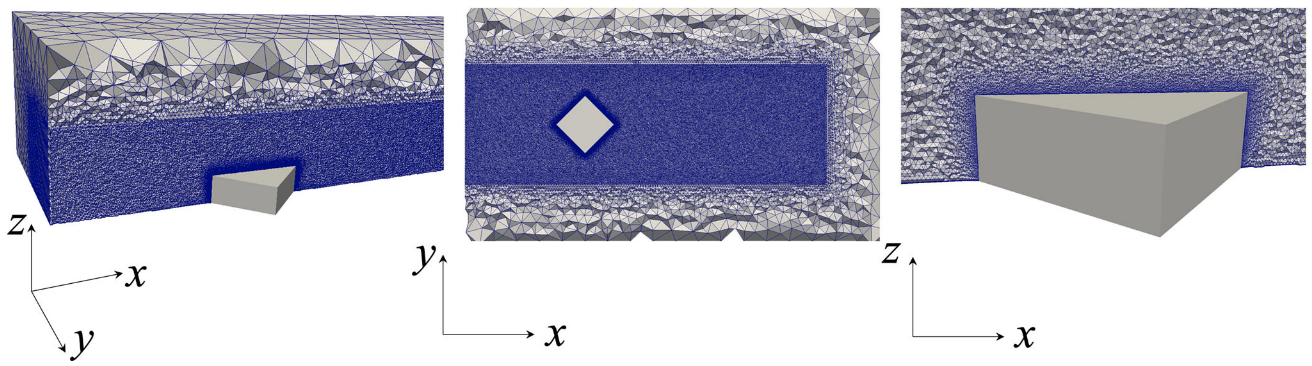
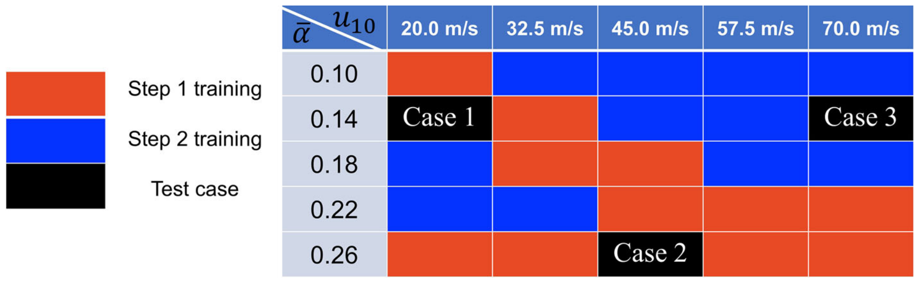


Fig. 3 The mesh utilized in the CFD simulations. Left: 3D view. Middle: x-y plane. Right: x-z plane



🟥	Step 1 training
🟦	Step 2 training
⬛	Test case

$\bar{\alpha}$ \ u_{10}	20.0 m/s	32.5 m/s	45.0 m/s	57.5 m/s	70.0 m/s
0.10	🟥	🟦	🟦	🟦	🟦
0.14	Case 1	🟥	🟦	🟦	Case 3
0.18	🟦	🟥	🟥	🟦	🟦
0.22	🟦	🟦	🟥	🟥	🟥
0.26	🟥	🟥	Case 2	🟥	🟥

Fig. 4 Data generation: Wind profiles considered for training and testing of the ML framework

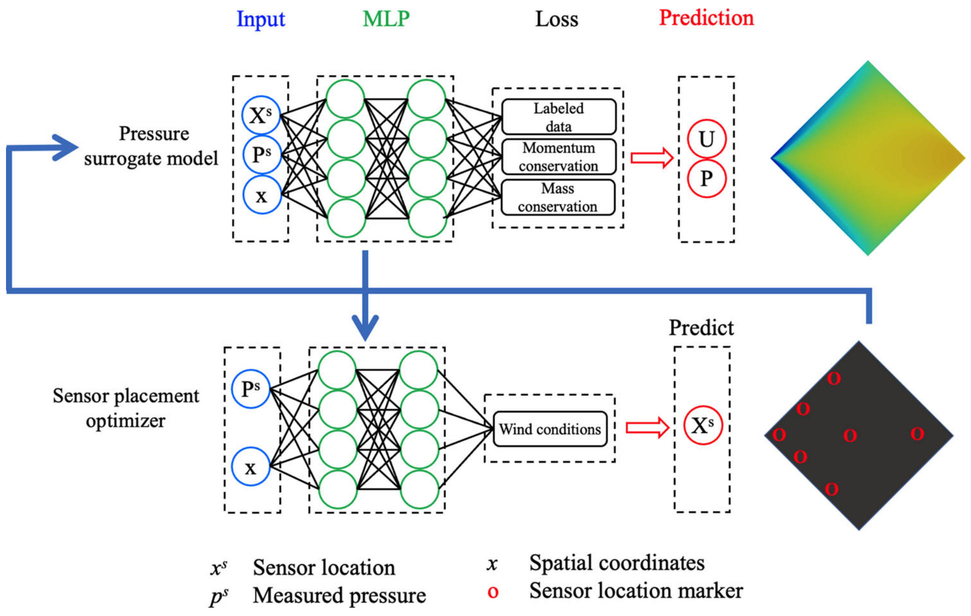


Fig. 5 ML framework for pressure prediction and sensor placement optimization

are the corresponding weights for each component. The data-driven component, L_{data}, which measures the discrepancy between the NN prediction and labeled measured pressure from sensors, is defined as

$$\mathcal{R}_{data} = \sum_{s=1}^{N_s} \left[p_{NN}(\boldsymbol{x}_s, \boldsymbol{x}_s; \boldsymbol{W}, \boldsymbol{b}) - \tilde{p}_s \right]^2 \tag{13}$$

where $\tilde{p}_s$ denotes the sensor pressure data and N_s denotes the number of sensors. Conventional off-the-shelf ML models only employ $\mathcal{R}_{data}$ in the training process, which relies on big data. Given the cost of data generation and the limited data available, the physics-informed components $\mathcal{R}_{pde}$ and $\mathcal{R}_{bc}$ are added to the objective function to alleviate the dependence on big-data. The physics are encoded into the NN by penalizing the residuals of the PDEs and associated BCs of the physical principles over a set of collocation points in the interior (denoted by Ω) and a set of collocation points on the boundary (denoted by $\partial\Omega$). For the problem considered in this paper, the physics are the RANS equations with turbulence closure model defined in Eqs. 4 and 5 and boundary conditions defined in Eq. 3. With the above definitions, $\mathcal{R}_{pde}$ and $\mathcal{R}_{bc}$ are specified as

$$\begin{aligned}\mathcal{R}_{pde} &= \left|\left|\ \mathcal{L}_M[\ \boldsymbol{u}_{NN}(\boldsymbol{x}, \boldsymbol{x}_s; \boldsymbol{W}, \boldsymbol{b}),\ \ p_{NN}(\boldsymbol{x}, \boldsymbol{x}_s; \boldsymbol{W}, \boldsymbol{b})\ ]\ \right|\right|^2_{\Omega} \\ &\quad + \left|\left|\ \mathcal{L}_C[\ \boldsymbol{u}_{NN}(\boldsymbol{x}, \boldsymbol{x}_s; \boldsymbol{W}, \boldsymbol{b}),\ \ p_{NN}(\boldsymbol{x}, \boldsymbol{x}_s; \boldsymbol{W}, \boldsymbol{b})\ ]\ \right|\right|^2_{\Omega} \quad (14)\\ \mathcal{R}_{bc} &= \left|\left|\ \mathcal{L}_{BC}[\ \boldsymbol{u}_{NN}(\boldsymbol{x}, \boldsymbol{x}_s; \boldsymbol{W}, \boldsymbol{b}),\ \ p_{NN}(\boldsymbol{x}, \boldsymbol{x}_s; \boldsymbol{W}, \boldsymbol{b})\ ]\ \right|\right|^2_{\partial\Omega} \quad (15)\end{aligned}$$

Once $\boldsymbol{W}^*$ and $\boldsymbol{b}^*$ are determined, the output of $\boldsymbol{u}_{NN}$ and $\boldsymbol{p}_{NN}$ can be easily achieved by a feed-forward evaluation of the NN with high efficiency since it merely costs a few matrix multiplications and vector additions.

4.2 Optimization of pressure sensor locations

In reality, a dense distribution of pressure sensors over the entire structural surface is infeasible and uneconomic. Besides, too many pressure sensors also introduce intrusive effects, leading to inaccurate pressure measurements. Thus, an out-layer NN is designed to optimize pressure sensor placement. The optimization algorithm aims to find the optimal sensor placement for the surrogate model that recovers the most accurate pressure profile over a wide range of wind conditions from a given number of pressure sensors. The sensor placement optimization is achieved by embedding the learned surrogate model obtained from the previous section in another data-driven neural network to minimize the overall discrepancies between surrogate model predictions and wind pressure by adjusting the sensor placement, namely,

$$\boldsymbol{X}^*_s = \underset{\boldsymbol{X}^s}{\operatorname{argmin}}\ \mathcal{L}_{loc}(\boldsymbol{X}^s) \quad (16)$$

where

$$\mathcal{L}_{loc}(\boldsymbol{X}^s) = \sum_{i=1}^{N_w} \left|\left|\boldsymbol{p}_{NN}(\boldsymbol{x}, \boldsymbol{X}_s, \tilde{P}_s; \boldsymbol{W^*}, \boldsymbol{b^*}) - \hat{\boldsymbol{p}}(\boldsymbol{x})\right|\right|^2_S \quad (17)$$

where S denotes the roof surface, $\boldsymbol{p}_{NN}$ is the full-field pressure prediction from the learned pressure surrogate model and $\hat{\boldsymbol{p}}(\boldsymbol{x})$ is the full-field prediction from the high-fidelity simulation. N_w denotes the number of wind conditions used for training the optimization algorithm. Once $\boldsymbol{X}^*_s$ is learned, the full-field pressure prediction with optimal pressure sensor placement can be obtained by

$$p^*_s(\boldsymbol{x}) = \boldsymbol{p}_{NN}(\boldsymbol{x}, \boldsymbol{X}^*_s, \tilde{P}_s; \boldsymbol{W^*}, \boldsymbol{b^*}) \quad (18)$$

4.2.1 Learning procedure

The PINN-based surrogate model and pressure sensor placement optimizer are trained by minimizing the loss functions defined in Eqs. 11 and 16. The minimization is executed by the following procedures: (1) The coordinates of collocation points and training data are substituted into the loss functions. (2) Take the derivatives of the loss functions with respect to the weight and bias in the neural network. (3) Update the weight and bias by a gradient descent. Most current machine learning frameworks solve the optimization problem by a stochastic gradient descent (SGD) algorithm, which is a stochastic approximation of the gradient descent optimization [97]. SGD only uses a subset of collocation points, randomly sampled from the input space at each iteration, to calculate the directional gradient. Research shows that SGD works very well to skip bad local minima. One issue with SGD is the oscillation of gradient direction caused by the random selection of sampled collocation points. In this paper, the Adam method that combines adaptive learning rate and momentum methods is used to improve convergence speed [98].

The PINN learning process needs the spatial and temporal derivatives with respect to the weight and bias, which can be accurately and efficiently calculated by using automatic differentiation (AD) [99]. The basic idea of AD is to use the chain rule to back-propagate derivatives from the output layer to the input layer since the connection between layers of a neural network is analytically defined. Compared to numerical differentiation techniques (e.g., finite difference and finite element), AD does not suffer from truncation or round-off errors, resulting in much higher accuracy. AD has been gaining increasing attention in the machine learning community and has been implemented in many modern deep learning frameworks, such as TensorFlow [100], PyTorch [101], Theano [102], and Caffe [103]. In this paper, the PINN-based surrogate and sensor placement optimizer are implemented in TensorFlow.

5 Applications

The performance of the proposed ML model depends on the number of pressure sensors and their locations. In this section, we investigate the effort of the number of pressure sensors on the predictive accuracy of the surrogate model for one specific wind condition. Based on the results, we then fix the number of pressure sensors and optimize their locations over various wind conditions.

5.1 Effet of number of pressure sensors on modeling accuracy for one wind condition

We consider the effects of different numbers on the accuracy of the ML model for particular wind conditions. 8, 16, and 32 sensors are chosen for investigation. To compare the accuracy of surrogate modeling using different sensor locations, for each number of sensors, we manually select two locations (see Fig. 6). One is highly concentrated in the interior (Placement 1), and another one is more uniformly distributed (Placement 2). We use test case 2 (see Fig. 4) to learn the optimized sensor placement for each number of pressure sensors. After training, the surrogate model takes the pressure information from sensor data as inputs to predict the full-field pressure. The recovered pressure fields from the surrogate model using Placement1, Placement 2, and the optimized placement are compared with the FEM result, which is treated as the ground truth.

Figure 7 shows the comparison of pressure prediction from the surrogate model with different numbers of sensors and sensor placements. The FEM results are plotted for comparison. We quantify the relative predictive error based on the difference from the FEM results and listed under each contour. For each number of sensors, the accuracy rank (from low to high) of sensor placements is Placement1, Placement2, and the Optimized placement, indicating the effectiveness and necessity of sensor location optimization. We find that, for each sensor placement, the pressure surrogate model's accuracy increases with more sensors. With a small number of sensors (e.g., eight sensors), the accuracy of all the placements is low. This means that more labeled data can enhance the accuracy of the ML model. However, installing more pressure sensors is not economically efficient. To have a well-balance of accuracy and cost, we choose 16 sensors in the next section and learn their optimal locations over a wide range of wind conditions.

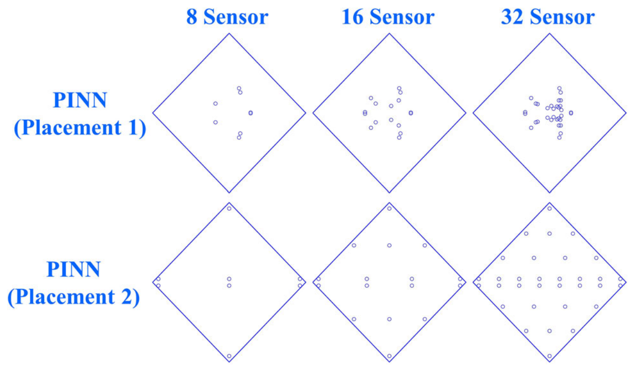


Fig. 6 Two manually selected sensor placements. Placement 1: Concentrated distribution. Placement 2: Uniform distribution

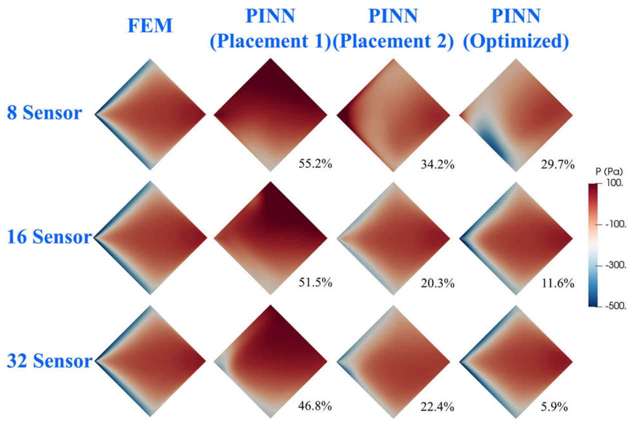


Fig. 7 Pressure prediction with different number of sensors (8, 16, and 32)

5.2 Learning optimal sensor placement for various wind conditions

With a fixed number of pressure sensors (16 in this case), we use the ML framework to learn their optimal placement that leads to the best surrogate modeling accuracy over various wind conditions. The FEM data from the 11 training wind cases defined in Fig. 4 is used to train the surrogate model. Then, another 11 cases are used to learn the sensor locations. The remaining three cases are used to test the performance of the learned optimal sensor placement.

Figure 8 shows the full-field pressure prediction of three testing cases. We also plot the results of the surrogate model using unoptimized sensor locations based on Placemen 1 and Placemen 2 (with 16 sensors) and the FEM results for comparison. With highly concentrated sensor distribution (Placement 1), the predictions from the pressure surrogate model fail to match the FEM results. The relative error with concentrated sensor distribution for Placement 1 is more than 25% for all three cases. The surrogate model with Placement 1 fails to capture the large pressure suction near the two front edges of the roof, which is widely seen in wind tun-

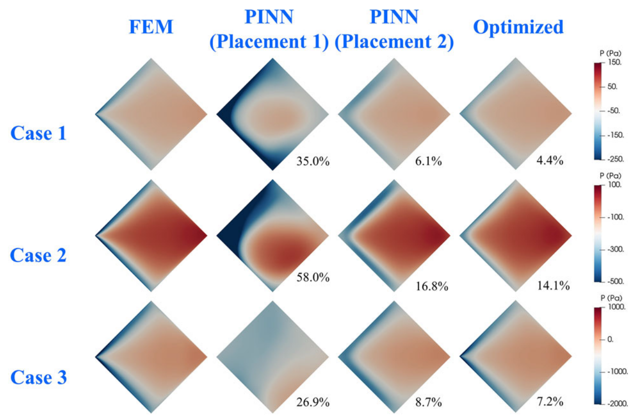


Fig. 8 Full-field pressure prediction of three testing cases

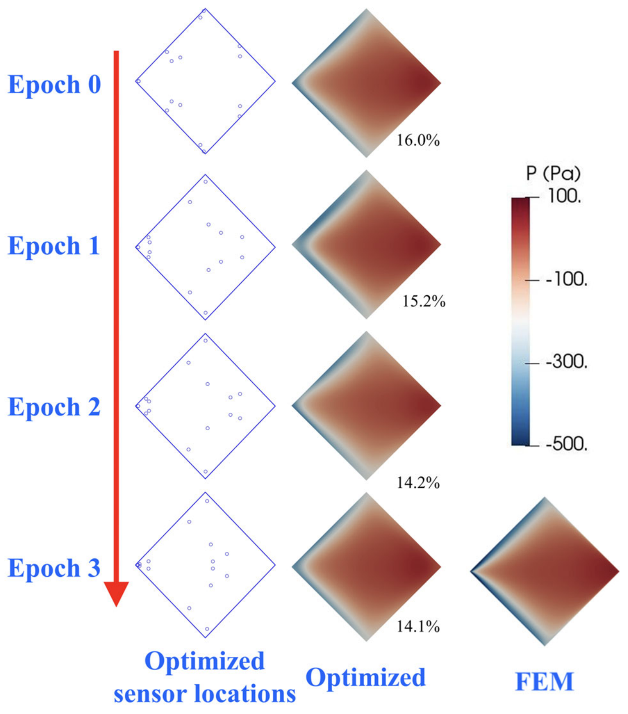


Fig. 9 Pressure sensor location for Case 2. The predicted pressure is obtained by substituting the pressure sensor locations into the surrogate model. Percentage represents the relative error of predictions with respect to the corresponding FEM results

nel experiments and the FEM results here. With Placement 2, the predicted accuracy is improved due to the uniform sensor distribution, indicating that the prediction error by the surrogate model is highly related to the pressure sensor locations. A biased sensor placement can lead to significant prediction errors. We find that the optimized placement provides higher accuracy than Placement 1 and Placement2 for all three cases, showing the value of the pressure placement optimizer.

To give some details of the optimization and show how better sensor location enhances predictive accuracy, we plot the dynamic evolution of the pressure sensor placement during the optimization process for Case 2 in Fig. 9. The corresponding pressure prediction from the surrogate model using these pressure sensor placements is plotted on the right. As the optimization goes on and reaches the optimal state, sensor placement gradually forms a symmetric configuration. Some sensors move towards the front edge to capture big pressure suction, and some move to the diagonal line to capture positive pressure. This process shows that the ML learned a sensor placement that tends to respect the symmetry of the problem and the underline pressure distribution pattern. Compared with the ground truth, one can see that the predictive accuracy has significantly improved with the optimized sensor placement compared with the initial randomly chosen sensor placement.

6 Conclusion

This paper presented an ML-based computational framework for surrogate modeling full-field wind pressure on civil structures and optimizing pressure sensor placement. The core of the framework is based on physics-informed machine learning, which combines physical principles and labeled data to train the ML models. We have shown the efficacy of the ML framework by deploying it to a classical flat roof subjected to single and multiple wind conditions. A careful assessment of the framework's performance was compared with standard FEM-based CFD simulations. We analyzed the effects of sensor numbers and sensor locations. Results showed that the ML framework maintains high accuracy and efficiency, showing great potential in digital twin and structural health monitoring applications. At this point, the framework can only predict the mean pressure profile. Future work will extend the framework to enable fast pressure prediction and pressure sensor placement optimization for the mean pressure and pressure fluctuation in a dynamic setting.

Acknowledgements The research was supported by the ZJU-UIUC Joint Research Center under the Grant of No. DREMES202001.

References

1. He Q, Barajas-Solano D, Tartakovsky G, Tartakovsky A (2020) Physics-informed neural networks for multiphysics data assimilation with application to subsurface transport. Adv Water Resour 141:103610
2. He Q, Tartakovsky A (2021) Physics-informed neural network method for forward and backward advection-dispersion equations. Water Resour Res 57(7):e2020WR029479
3. Liu Z, Bessa M, Liu W (2016) Self-consistent clustering analysis: an efficient multi-scale scheme for inelastic heterogeneous materials. Comput Methods Appl Mech Eng 306:319–341
4. Balu A, Nallagonda S, Xu F, Krishnamurthy A, Hsu M, Sarkar S (2019) A deep learning framework for design and analysis of surgical bioprosthetic heart valves. Sci Rep 9(1):1–12
5. Mozaffar M, Bostanabad R, Chen W, Ehmann K, Cao J, Bessa M (2019) Deep learning predicts path-dependent plasticity. Proc Natl Acad Sci 116(52):26414–26420
6. Liu Z (2020) Deep material network with cohesive layers: multistage training and interfacial failure analysis. Comput Methods Appl Mech Eng 363:112913
7. He Q, Tartakovsky A (2021) Physics-informed neural network method for forward and backward advection-dispersion equations. Water Resour Res 57(7):e2020WR029479
8. Yan W, Lin S, Kafka O, Lian Y, Yu C, Liu Z, Yan J, Wolff S, Wu H, Ndip-Agbor E, Mozaffar M, Ehmann K, Cao J, Wagner G, Liu W (2018) Data-driven multi-scale multi-physics models to derive process-structure-property relationships for additive manufacturing. Comput Mech 61:521–541
9. Paul A, Mozaffar M, Yang Z, Liao W, Choudhary A, Cao J, Agrawal A (2019) A real-time iterative machine learning approach for temperature profile prediction in additive manufacturing processes. In: 2019 IEEE International Conference on Data Science and Advanced Analytics (DSAA). IEEE, pp 541–550

10. Zhu Q, Liu Z, Yan J (2021) Machine learning for metal additive manufacturing: predicting temperature and melt pool fluid dynamics using physics-informed neural networks. Comput Mech 67(2):619–635
11. Xie X, Bennett J, Saha S, Lu Y, Cao J, Liu W, Gan Z (2021) Mechanistic data-driven prediction of as-built mechanical properties in metal additive manufacturing. npj Comput Mater 7(1):1–12
12. Zhao Z, Stuebner M, Lua J, Phan N, Yan J (2022) Full-field temperature recovery during water quenching processes via physics-informed machine learning. J Mater Process Technol 303:117534
13. Kozjek D, Carter FMI, Porter C, Mogonye J, Ehmann K, Cao J (2022) Data-driven prediction of next-layer melt pool temperatures in laser powder bed fusion based on co-axial high-resolution Planck thermometry measurements. J Manuf Process 79:81–90
14. Mozaffar M, Liao S, Xie X, Saha S, Park C, Cao J, Liu W, Gan Z (2021) Mechanistic artificial intelligence (mechanistic-ai) for modeling, design, and control of advanced manufacturing processes: current state and perspectives. J Mater Process Technol 117485
15. Jiang Z, Ehmann K, Cao J (2022) Prediction of forming temperature in electrically-assisted double-sided incremental forming using a neural network. J Mater Process Technol 302:117486
16. Zhang L, Lu Y, Tang S, Liu W (2022) Hidenn-td: reduced-order hierarchical deep learning neural networks. Comput Methods Appl Mech Eng 389:114414
17. Kats D, Wang Z, Gan Z, Liu W, Wagner G, Lian Y (2022) A physics-informed machine learning method for predicting grain structure characteristics in directed energy deposition. Comput Mater Sci 202:110958
18. Sun L, Gao H, Pan S, Wang J (2020) Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data. Comput Methods Appl Mech Eng 361:112732
19. Hughes TJR, Mallet M (1986) A new finite element formulation for computational fluid dynamics: III. The generalized streamline operator for multidimensional advective–diffusive systems. Comput Methods Appl Mech Eng 58:305–328
20. Tezduyar TE (1992) Stabilized finite element formulations for incompressible flow computations. Adv Appl Mech 28:1–44. https://doi.org/10.1016/S0065-2156(08)70153-4
21. Tezduyar T, Sathe S (2003) Stabilization parameters in SUPG and PSPG formulations. J Comput Appl Mech 4:71–88
22. Takizawa K, Tezduyar TE (2012) Space-time fluid-structure interaction methods. Math Models Methods Appl Sci 22(supp02):1230001. https://doi.org/10.1142/S0218202512300013
23. Takizawa K, Tezduyar TE, Kuraishi T (2015) Multiscale ST methods for thermo-fluid analysis of a ground vehicle and its tires. Math Models Methods Appl Sci 25:2227–2255. https://doi.org/10.1142/S0218202515400072
24. Takizawa K, Tezduyar TE (2011) Multiscale space-time fluid-structure interaction techniques. Comput Mech 48:247–267. https://doi.org/10.1007/s00466-011-0571-z
25. Bazilevs Y, Calo VM, Hughes TJR, Zhang Y (2008) Isogeometric fluid-structure interaction: theory, algorithms, and computations. Comput Mech 43:3–37
26. Takizawa K, Bazilevs Y, Tezduyar TE (2012) Space-time and ALE-VMS techniques for patient-specific cardiovascular fluid-structure interaction modeling. Arch Comput Methods Eng 19:171–225. https://doi.org/10.1007/s11831-012-9071-3
27. Bazilevs Y, Hsu M-C, Takizawa K, Tezduyar TE (2012) ALE-VMS and ST-VMS methods for computer modeling of wind-turbine rotor aerodynamics and fluid-structure interaction. Math Models Methods Appl Sci 22(supp02):1230002. https://doi.org/10.1142/S0218202512300025
28. Bazilevs Y, Takizawa K, Tezduyar TE (2013) Computational fluid-structure interaction: methods and applications. Wiley, New York. https://doi.org/10.1002/9781118483565
29. Bazilevs Y, Takizawa K, Tezduyar TE (2013) Challenges and directions in computational fluid-structure interaction. Math Models Methods Appl Sci 23:215–221. https://doi.org/10.1142/S0218202513400010
30. Bazilevs Y, Takizawa K, Tezduyar TE (2015) New directions and challenging computations in fluid dynamics modeling with stabilized and multiscale methods. Math Models Methods Appl Sci 25:2217–2226. https://doi.org/10.1142/S0218202515020029
31. Bazilevs Y, Takizawa K, Tezduyar TE (2019) Computational analysis methods for complex unsteady flow problems. Math Models Methods Appl Sci 29:825–838. https://doi.org/10.1142/S0218202519020020
32. Calderer R, Zhu L, Gibson R, Masud A (2015) Residual-based turbulence models and arbitrary Lagrangian–Eulerian framework for free surface flows. Math Models Methods Appl Sci 25(12):2287–2317
33. Takizawa K, Tezduyar TE (2012) Computational methods for parachute fluid-structure interactions. Arch Comput Methods Eng 19:125–169. https://doi.org/10.1007/s11831-012-9070-4
34. Takizawa K, Fritze M, Montes D, Spielman T, Tezduyar TE (2012) Fluid-structure interaction modeling of ringsail parachutes with disreefing and modified geometric porosity. Comput Mech 50:835–854. https://doi.org/10.1007/s00466-012-0761-3
35. Takizawa K, Tezduyar TE, Boben J, Kostov N, Boswell C, Buscher A (2013) Fluid-structure interaction modeling of clusters of spacecraft parachutes with modified geometric porosity. Comput Mech 52:1351–1364. https://doi.org/10.1007/s00466-013-0880-5
36. Takizawa K, Tezduyar TE, Boswell C, Tsutsui Y, Montel K (2015) Special methods for aerodynamic-moment calculations from parachute FSI modeling. Comput Mech 55:1059–1069. https://doi.org/10.1007/s00466-014-1074-5
37. Kalro V, Tezduyar TE (2000) A parallel 3D computational method for fluid-structure interactions in parachute systems. Comput Methods Appl Mech Eng 190:321–332. https://doi.org/10.1016/S0045-7825(00)00204-8
38. Zhu Q, Yan J, Tejada-Martínez A, Bazilevs Y (2020) Variational multiscale modeling of Langmuir turbulent boundary layers in shallow water using isogeometric analysis. Mech Res Commun 108:103570. https://doi.org/10.1016/j.mechrescom.2020.103570
39. Ravensbergen M, Helgedagsrud TA, Bazilevs Y, Korobenko A (2020) A variational multiscale framework for atmospheric turbulent flows over complex environmental terrains. Comput Methods Appl Mech Eng 368:113182. https://doi.org/10.1016/j.cma.2020.113182
40. Yan J, Korobenko A, Tejada-Martinez AE, Golshan R, Bazilevs Y (2017) A new variational multiscale formulation for stratified incompressible turbulent flows. Comput Fluids 158:150–156. https://doi.org/10.1016/j.compfluid.2016.12.004
41. Cen H, Zhou Q, Korobenko A (2022) Wall-function-based weak imposition of Dirichlet boundary condition for stratified turbulent flows. Comput Fluids 234:105257
42. Bazilevs Y, Hsu M-C, Akkerman I, Wright S, Takizawa K, Henicke B, Spielman T, Tezduyar TE (2011) 3D simulation of wind turbine rotors at full scale. Part I: geometry modeling and aerodynamics. Int J Numer Methods Fluids 65:207–235. https://doi.org/10.1002/fld.2400
43. Takizawa K, Henicke B, Tezduyar TE, Hsu M-C, Bazilevs Y (2011) Stabilized space-time computation of wind-turbine rotor aerodynamics. Comput Mech 48:333–344. https://doi.org/10.1007/s00466-011-0589-2
44. Takizawa K, Henicke B, Montes D, Tezduyar TE, Hsu M-C, Bazilevs Y (2011) Numerical-performance studies for the

stabilized space-time computation of wind-turbine rotor aerodynamics. Comput Mech 48:647–657. https://doi.org/10.1007/s00466-011-0614-5
45. Takizawa K, Tezduyar TE, McIntyre S, Kostov N, Kolesar R, Habluetzel C (2014) Space-time VMS computation of wind-turbine rotor and tower aerodynamics. Comput Mech 53:1–15. https://doi.org/10.1007/s00466-013-0888-x
46. Takizawa K, Bazilevs Y, Tezduyar TE, Hsu M-C, Øiseth O, Mathisen KM, Kostov N, McIntyre S (2014) Engineering analysis and design with ALE-VMS and space-time methods. Arch Comput Methods Eng 21:481–508. https://doi.org/10.1007/s11831-014-9113-0
47. Takizawa K (2014) Computational engineering analysis with the new-generation space-time methods. Comput Mech 54:193–211. https://doi.org/10.1007/s00466-014-0999-z
48. Bazilevs Y, Takizawa K, Tezduyar TE, Hsu M-C, Kostov N, McIntyre S (2014) Aerodynamic and FSI analysis of wind turbines with the ALE-VMS and ST-VMS methods. Arch Comput Methods Eng 21:359–398. https://doi.org/10.1007/s11831-014-9119-7
49. Takizawa K, Tezduyar TE, Mochizuki H, Hattori H, Mei S, Pan L, Montel K (2015) Space-time VMS method for flow computations with slip interfaces (ST-SI). Math Models Methods Appl Sci 25:2377–2406. https://doi.org/10.1142/S0218202515400126
50. Otoguro Y, Mochizuki H, Takizawa K, Tezduyar TE (2020) Space-time variational multiscale isogeometric analysis of a tsunami-shelter vertical-axis wind turbine. Comput Mech 66:1443–1460. https://doi.org/10.1007/s00466-020-01910-5
51. Ravensbergen M, Bayram AM, Korobenko A (2020) The actuator line method for wind turbine modelling applied in a variational multiscale framework. Comput Fluids 201:104465. https://doi.org/10.1016/j.compfluid.2020.104465
52. Korobenko A, Hsu M-C, Akkerman I, Bazilevs Y (2013) Aerodynamic simulation of vertical-axis wind turbines. J Appl Mech 81:021011. https://doi.org/10.1115/1.4024415
53. Bazilevs Y, Korobenko A, Deng X, Yan J, Kinzel M, Dabiri JO (2014) FSI modeling of vertical-axis wind turbines. J Appl Mech 81:081006. https://doi.org/10.1115/1.4027466
54. Korobenko A, Bazilevs Y, Takizawa K, Tezduyar TE (2018) Recent advances in ALE-VMS and ST-VMS computational aerodynamic and FSI analysis of wind turbines. In: Tezduyar TE (ed) Frontiers in computational fluid-structure interaction and flow simulation: research from lead investigators under forty - 2018, modeling and simulation in science, engineering and technology. Berlin, Springer, pp 253–336. https://doi.org/10.1007/978-3-319-96469-0_7
55. Korobenko A, Bazilevs Y, Takizawa K, Tezduyar TE (2019) Computer modeling of wind turbines: 1. ALE-VMS and ST-VMS aerodynamic and FSI analysis. Arch Comput Methods Eng 26:1059–1099. https://doi.org/10.1007/s11831-018-9292-1
56. Bayram AM, Bear C, Bear M, Korobenko A (2020) Performance analysis of two vertical-axis hydrokinetic turbines using variational multiscale method. Comput Fluids 200:104432. https://doi.org/10.1016/j.compfluid.2020.104432
57. Yan J, Korobenko A, Deng X, Bazilevs Y (2016) Computational free-surface fluid-structure interaction with application to floating offshore wind turbines. Comput Fluids 141:155–174. https://doi.org/10.1016/j.compfluid.2016.03.008
58. Yan J, Deng X, Xu F, Xu S, Zhu Q (2020) Numerical simulations of two back-to-back horizontal axis tidal stream turbines in free-surface flows. J Appl Mech. https://doi.org/10.1115/1.4046317
59. Kuraishi T, Zhang F, Takizawa K, Tezduyar TE (2021) Wind turbine wake computation with the ST-VMS method, isogeometric discretization and multidomain method: I. Computational framework. Comput Mech 68(1):113–130
60. Kuraishi T, Zhang F, Takizawa K, Tezduyar TE (2021) Wind turbine wake computation with the ST-VMS method, isogeometric discretization and multidomain method: II. Spatial and temporal resolution. Comput Mech 68(1):175–184
61. Ravensbergen M, Mohamed A, Korobenko A (2020) The actuator line method for wind turbine modelling applied in a variational multiscale framework. Comput Fluids 201:104465
62. Mohamed A, Bear C, Bear M, Korobenko A (2020) Performance analysis of two vertical-axis hydrokinetic turbines using variational multiscale method. Comput Fluids 200:104432
63. Bayram A, Korobenko A (2020) Variational multiscale framework for cavitating flows. Comput Mech 66:1–19
64. Yan J, Deng X, Korobenko A, Bazilevs Y (2017) Free-surface flow modeling and simulation of horizontal-axis tidal-stream turbines. Comput Fluids 158:157–166. https://doi.org/10.1016/j.compfluid.2016.06.016
65. Zhu Q, Yan J (2021) A moving-domain CFD solver in FEniCS with applications to tidal turbine simulations in turbulent flows. Comput Math Appl 81:532–546
66. Bayram AM, Korobenko A (2020) Variational multiscale framework for cavitating flows. Comput Mech 66:49–67. https://doi.org/10.1007/s00466-020-01840-2
67. Bayram A, Korobenko A (2020) Variational multiscale framework for cavitating flows. Comput Mech 66(1):49–67
68. Codoni D, Moutsanidis G, Hsu M-C, Bazilevs Y, Johansen C, Korobenko A (2021) Stabilized methods for high-speed compressible flows: toward hypersonic simulations. Comput Mech 67:785–809. https://doi.org/10.1007/s00466-020-01963-6
69. Terahara T, Takizawa K, Tezduyar TE, Bazilevs Y, Hsu M-C (2020) Heart valve isogeometric sequentially-coupled FSI analysis with the space-time topology change method. Comput Mech 65:1167–1187. https://doi.org/10.1007/s00466-019-01813-0
70. Hsu M-C, Kamensky D, Bazilevs Y, Sacks MS, Hughes TJR (2014) Fluid-structure interaction analysis of bioprosthetic heart valves: significance of arterial wall deformation. Comput Mech 54:1055–1071. https://doi.org/10.1007/s00466-014-1059-4
71. Johnson EL, Wu MCH, Xu F, Wiese NM, Rajanna MR, Herrema AJ, Ganapathysubramanian B, Hughes TJR, Sacks MS, Hsu M-C (2020) Thinner biological tissues induce leaflet flutter in aortic heart valve replacements. Proc Natl Acad Sci 117:19007–19016
72. Takizawa K, Bazilevs Y, Tezduyar TE, Hsu M-C (2019) Computational cardiovascular flow analysis with the variational multiscale methods. J Adv Eng Comput 3:366–405. https://doi.org/10.25073/jaec.201932.245
73. Kuraishi T, Terahara T, Takizawa K, Tezduyar T (2022) Computational flow analysis with boundary layer and contact representation: I. Tire aerodynamics with road contact. J Mech 38:77–87
74. Terahara T, Kuraishi T, Takizawa K, Tezduyar T (2022) Computational flow analysis with boundary layer and contact representation: II. Heart valve flow with leaflet contact. J Mech 38:185–194
75. Otoguro Y, Takizawa K, Tezduyar TE, Nagaoka K, Avsar R, Zhang Y (2019) Space-time VMS flow analysis of a turbocharger turbine with isogeometric discretization: aomputations with time-dependent and steady-inflow representations of the intake/exhaust cycle. Comput Mech 64:1403–1419. https://doi.org/10.1007/s00466-019-01722-2
76. Otoguro Y, Takizawa K, Tezduyar TE, Nagaoka K, Mei S (2019) Turbocharger turbine and exhaust manifold flow computation with the space-time variational multiscale method and isogeometric analysis. Comput Fluids 179:764–776. https://doi.org/10.1016/j.compfluid.2018.05.019
77. Xu F, Moutsanidis G, Kamensky D, Hsu M-C, Murugan M, Ghoshal A, Bazilevs Y (2017) Compressible flows on moving domains: stabilized methods, weakly enforced essential boundary conditions, sliding interfaces, and application to gas-turbine modeling. Comput Fluids 158:201–220. https://doi.org/10.1016/j.compfluid.2017.02.006

78. Kuraishi T, Takizawa K, Tabata S, Asada S, Tezduyar TE (2014) Multiscale thermo-fluid analysis of a tire. In: Proceedings of the 19th Japan society of computational engineering and science conference, Hiroshima, Japan
79. Takizawa K, Tezduyar TE, Kuraishi T (2016) Flow analysis around a tire with actual geometry, road contact and deformation. in preparation
80. Kuraishi T, Takizawa K, Tezduyar TE (2018) Space-time computational analysis of tire aerodynamics with actual geometry, road contact and tire deformation. In: Tezduyar TE (ed) Frontiers in computational fluid-structure interaction and flow simulation: research from lead investigators under forty—2018. Modeling and simulation in science, engineering and technology. Springer, Berlin, pp 337–376. https://doi.org/10.1007/978-3-319-96469-0_8
81. Kuraishi T, Takizawa K, Tezduyar TE (2019) Tire aerodynamics with actual tire geometry, road contact and tire deformation. Comput Mech 63:1165–1185. https://doi.org/10.1007/s00466-018-1642-1
82. Kuraishi T, Takizawa K, Tezduyar TE (2019) Space-time computational analysis of tire aerodynamics with actual geometry, road contact, tire deformation, road roughness and fluid film. Comput Mech 64:1699–1718. https://doi.org/10.1007/s00466-019-01746-8
83. Saad Y, Schultz M (1986) GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J Sci Stat Comput 7:856–869
84. Holmes J (2007) Wind loading of structures. CRC Press, Berlin
85. Mooneghi M, Irwin P, Chowdhury A (2014) Large-scale testing on wind uplift of roof pavers. J Wind Eng Ind Aerodyn 128:22–36
86. Pantua CAJ, Calautit JK, Wu Y (2021) Sustainability and structural resilience of building integrated photovoltaics subjected to typhoon strength winds. Appl Energy 301:117437
87. Kind R (1986) Worst suctions near edges of flat rooftops on low-rise buildings. J Wind Eng Ind Aerodyn 25(1):31–47
88. Raissi M, Perdikaris P, Karniadakis GE (2019) Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys 378:686–707
89. Liu W, Wang Z, Liu X, Zeng N, Liu Y, Alsaadi F (2017) A survey of deep neural network architectures and their applications. Neurocomputing 234:11–26
90. Schwing AG, Urtasun R. Fully connected deep structured networks. arXiv:1503.02351
91. Lawrence S, Giles CL, Tsoi AC, Back AD (1997) Face recognition: a convolutional neural-network approach. IEEE Trans Neural Netw 8(1):98–113
92. Mikolov T, Karafiát M, Burget L, Cernockỳ J, Khudanpur S (2010) Recurrent neural network based language model. In: Interspeech, vol 2. Makuhari, pp 1045–1048
93. Liang S, Srikant R. Why deep neural networks for function approximation? arXiv:1610.04161
94. Sibi P, Jones SA, Siddarth P (2013) Analysis of different activation functions using back propagation neural networks. J Theor Appl Inf Technol 47(3):1264–1268
95. Maas A, Hannun A, Ng A (2013) Rectifier nonlinearities improve neural network acoustic models. In: Proceedings of ICML, vol 30, p 3
96. Eger S, Youssef P, Gurevych I. Is it time to swish? comparing deep learning activation functions across nlp tasks. arXiv:1901.02671
97. Ruder S. An overview of gradient descent optimization algorithms. arXiv:1609.04747
98. Kingma DP, Ba J. Adam: a method for stochastic optimization. arXiv:1412.6980
99. Griewank A et al (1989) On automatic differentiation. Math Program Recent Dev Appl 6(6):83–107
100. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M et al. Tensorflow: large-scale machine learning on heterogeneous distributed systems. arXiv:1603.04467
101. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L et al. Pytorch: an imperative style, high-performance deep learning library. In: Advances in neural information processing systems, vol 32
102. Bastien F, Lamblin P, Pascanu R, Bergstra J, Goodfellow I, Bergeron A, Bouchard N, Warde-Farley D, Bengio Y. Theano: new features and speed improvements. arXiv:1211.5590
103. Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T (2014) Caffe: convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM international conference on Multimedia, pp 675–678