The 9th European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS Congress 2024
3–7 June 2024, Lisboa, Portugal

CONVOLUTION NEURAL NETWORK FOR FLUID FLOW SIMULATIONS IN CASCADE WITH OSCILLATING BLADES

Václav Heidler[1], Ondřej Bublík[1] and Jan Vimmr[1,2]

[1] NTIS - New Technology for the Information Society, Faculty of Applied Sciences, University of West Bohemia, Technická 8, 30100 Pilsen, Czech Republic
[2] Department of Mechanics, Faculty of Applied Sciences, University of West Bohemia, Technická 8, 30100 Pilsen, Czech Republic

Key words: convolution neural network, compressible fluid flow, blade cascade, fluid-structure interaction

Summary. The analysis of blade cascades for flutter onset is crucial for the operational safety of turbomachinery. Despite significant advancements in computational power, comprehensive CFD analysis remains a time-consuming task. This paper aims to mitigate the time demands of CFD calculations by utilizing convolutional neural networks (CNNs). The developed model is tested on a Fluid-Structure Interaction (FSI) analysis, revealing the conditions for flutter.

1 INTRODUCTION

Recent advances in deep learning have significantly impacted the field of computational fluid dynamics (CFD), especially through the application of convolutional neural networks (CNNs). These networks are capable of modeling flow characteristics, handling complex geometries, and processing extensive datasets, making them invaluable for fluid flow predictions. Their ability to learn from raw data and make real-time predictions has established CNNs as powerful tools in various applications of fluid dynamics. Pioneering work [1] marked a significant milestone by applying CNNs to predict steady fluid flow. In work [8] authors extended this approach to predict unsteady turbulent fluid flows, inspiring numerous subsequent studies exploring CNN applications in fluid dynamics, including works [10, 2, 3, 4].

This study aims to utilize a convolutional neural network to predict unsteady flow fields in a cascade of oscillating blades. The primary motivation for using CNNs is their high prediction speed, which addresses the significant time constraints of traditional CFD analyses. The developed neural network model is validated against the FlowPro CFD solver [6] in an unsteady 2D problem with a moving boundary. FlowPro is based on the Discontinuous Galerkin finite element method and was also used for the generation of the training dataset for the model. Further testing is conducted on an FSI task to reveal the conditions for flutter onset. The usage of CNN in FSI analysis not only reduces the computational cost but also opens new avenues for real-time and large-scale applications.

2 PROBLEM DESCRIPTION

In this work, we examine a system of bodies composed of three blades. Specifically, we focus on solving a cross-section of the blade cascade, as the problem is defined in a 2D space. The

motion of the first and third blades is prescribed by a harmonic function, as follows

$$y_1 = A\sin(2\pi f + \Phi), \quad y_3 = A\sin(2\pi f - \Phi),$$

where A is the amplitude, f is the frequency and Φ is the phase angle. The middle blade is free to move with one degree of freedom along the yy-axis, and its motion is governed by a mass-spring model.

$$m\ddot{y}_2 + ky_2 = L,$$

where m is the blade mass, k is the blade stiffness and L is the lift force from the fluid, defined using the flow field variables p, u, and v as follows:

$$L = \oint_\Gamma \left(pn_y - \frac{1}{Re}\left[\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right) n_x + 2\frac{\partial v}{\partial y} n_y \right] \right) dl.$$

The situation is illustrated in Fig. 1. The stiffness k of the blade is chosen such that the natural

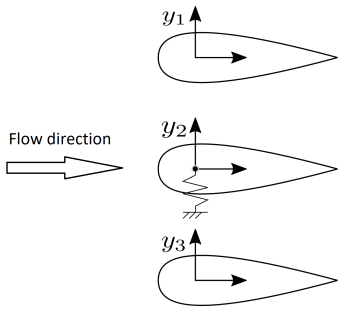

Figure 1: Problem setup

frequency of the oscillating blade matches the prescribed frequency f of the oscillating blades, i.e., $k = m(2\pi f)^2$.

2.1 Mesh generation and deformation

For the given problem, we consider a structured computational grid of size 128×128 points generated by an elliptic grid generator; see Fig. 3. The grid points entering the model vary depending on the position of the blades. To compute the deformed grid, we use a simple algorithm based on blending functions, which applies to computing grid deformation for a system of rigid bodies. For each of the blades, we first precompute the so-called blending function $\mathrm{bf}^b(\boldsymbol{x})$ as a solution to the Laplace equation with the following Dirichlet boundary conditions

$$\mathrm{bf}^b(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \boldsymbol{x} \in \Gamma_j, \quad b = j, \\ 0 & \text{if } \boldsymbol{x} \in \Gamma_j, \quad b \neq j \end{cases}$$

where Γ_j is the boundary of j-th blade.

Let $\boldsymbol{x}^0_{ij} = [x^0_{ij}, y^0_{ij}]$ are the points of the initial undeformed mesh. Let $\boldsymbol{x}^b_t = [x^b, y^b]$ is the translation vector of a blade b. Then we can compute the points of the deformed mesh as follows

$$\boldsymbol{x}_{ij} = \boldsymbol{x}^0_{ij} + \sum_b \mathrm{bf}^b_{ij} \boldsymbol{x}^b_t$$

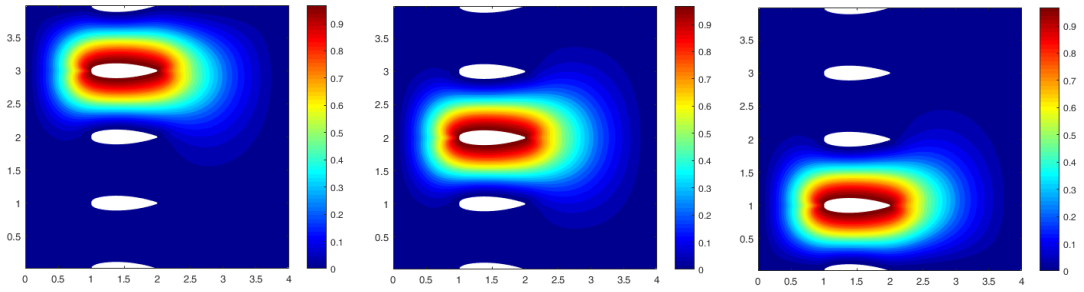

Figure 2: Blending functions. The blue lines shows initial positions of blades.

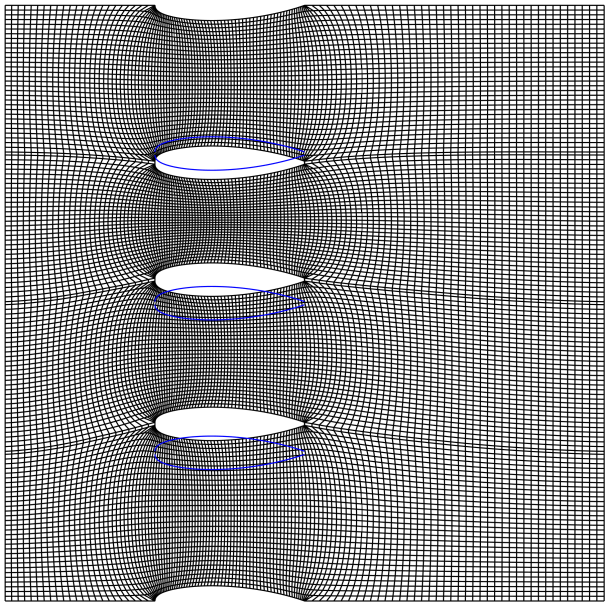

Figure 3: Illustration of the Deformed State of the Mesh: The Blue Line Represents the Undeformed Position of the Profiles.

3 NEURAL NETWORK MODEL

To predict unsteady incompressible fluid flow in blade cascade, we use a specialized convolutional encoder-decoder neural network known as U-Net [5], as depicted in Fig. 4. The encoder compresses input data from the initial layer into an encoded representation by applying convolutional filters followed by pooling layers, progressively down-sampling the data until it is sufficiently compressed. The decoder, which mirrors the encoder, then increases the spatial resolution and reduces the number of channels until the desired output is reconstructed. This process involves upsampling and deconvolution, the inverse operations of pooling and convolution. The architecture also includes skip connections, which concatenate each encoder layer with its corresponding decoder layer. These connections ensure that partially encoded input information from each encoder layer is available during output reconstruction. Consequently, the number of input channels in the decoder is increased by the number of channels in the encoder due to this concatenation. The neural network's input is a three-dimensional array with

dimensions $128 \times 128 \times 8$, containing eight values for each grid point: the x and y coordinates of the grid points, corresponding boundary mark, mesh velocities $\dot{x}$ and $\dot{y}$ and flow field u, v and p at time level t^n. The boundary mark is a binary value, equal to 1 if the grid point is on the boundary of the blade profile, and 0 if the grid point is within the fluid domain. The network outputs the pressure and velocity fields at the grid points at time t^{n+1}. The key parameters of the developed CNN structure are summarized in Table 1.

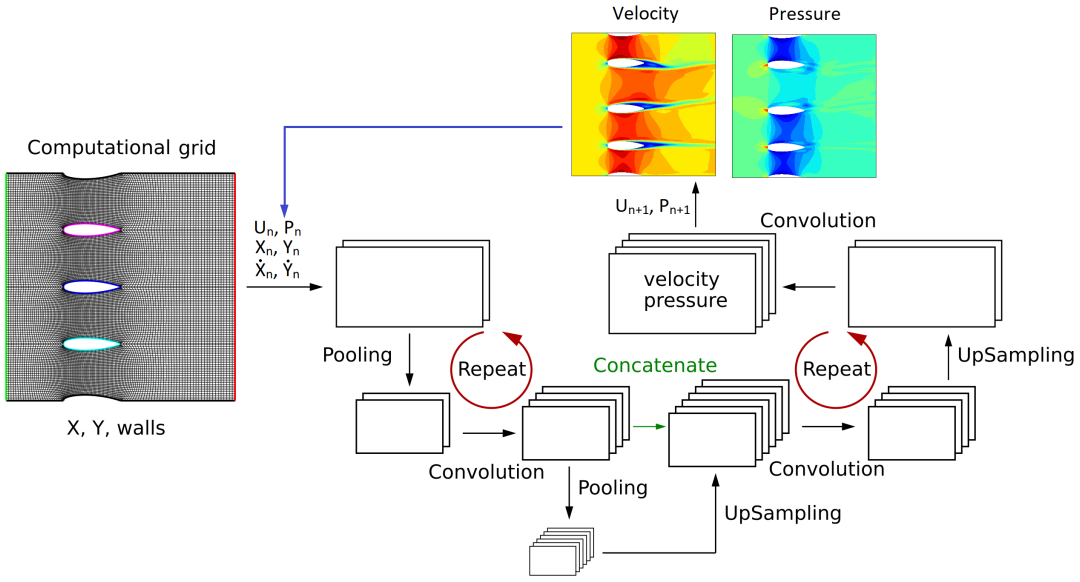

Figure 4: Neural network architekture

Table 1: Hyperparameters of the U-Net

Parameter	Value
Input shape	$128 \times 128 \times 8$
Output shape	$128 \times 128 \times 3$
Initial number of filters	12
Convolution frame	3×3
Pooling frame	2×2
Pooling function	Max pooling
Activation function	ReLU
Activation function at output	Sigmoid
Depth	7
Trainable parameters	1,062,039

4 MODEL TRAINING AND VALIDATION

The proposed neural network model was trained using data from four CFD simulations. In each simulation, the harmonic kinematic motion of the three blades was randomly set as

$y_i = A_i \sin(2\pi f_i)$, $i = 1, 2, 3$. The amplitudes A_i and frequencies f_i were randomly chosen from the intervals $A \in [0.05, 0.2]$ and $f \in [0.1, 2.0]$. The computation of the flow field was performed using CFD software based on the discontinuous Galerkin finite element method [6]. The trained model was validated on a random kinematic motion, see Fig. 6, where the lift force acting on each blade over time is compared. The CFD calculation is shown in blue, and the neural network prediction is shown in orange.

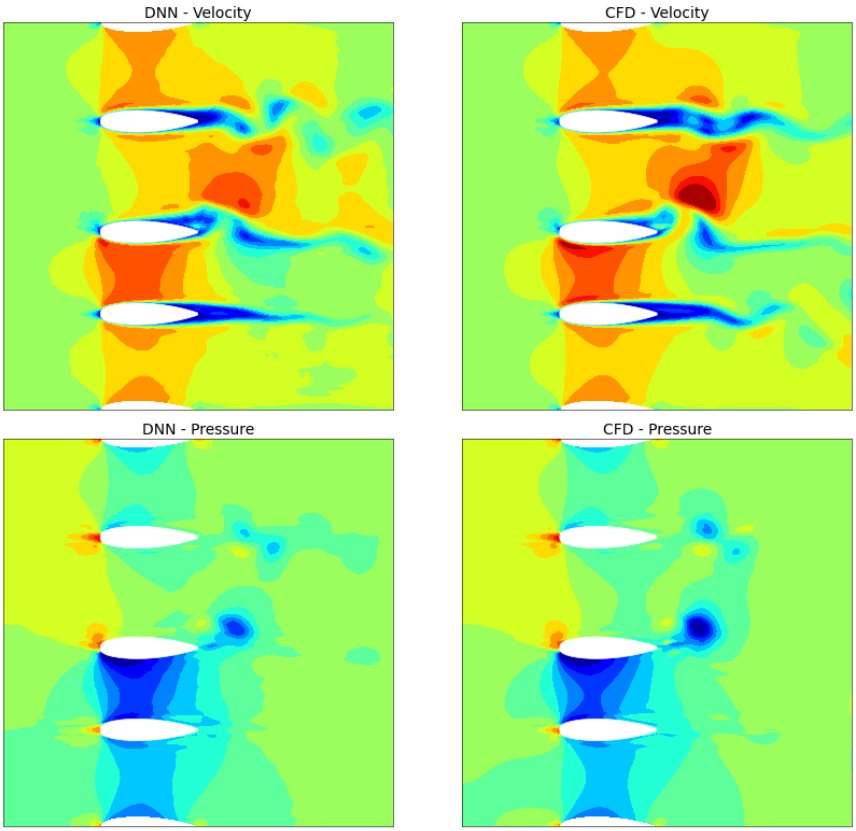

Figure 5: Comparison of the flow field predicted by the neural network (left) and the flow field computed using CFD simulation (right).

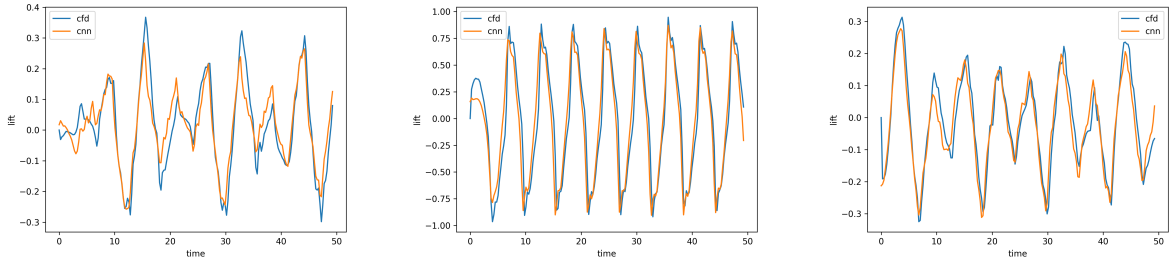

Figure 6: Comparison of lift between CFD simulation (blue) and neural network prediction (orange) for the upper (left), middle (middle), and lower (right) blade.

5 FSI SIMULATION RESULTS

Consider the FSI problem described in the introductory section. For the upper and lower blades, we set a harmonic motion with an amplitude of $A = 0.15$ and a frequency of $f = 0.1$. The parameters for the middle blade are chosen as follows. The natural frequency is considered the same as the oscillation frequency of the surrounding blades, i.e., $f_n = f = 0.1$. For the chosen mass $m = 20$, the spring stiffness is calculated as $k = m(2\pi f)^2 = 7.84$. The goal of this study is to observe the excitation of the middle blade depending on the phase angle Φ. We performed a total of four simulations for the fundamental phase angles $\Phi = -\pi/2, -\pi/4, 0, \pi/4$. Figs.7-10 show the lift forces acting on the blades and the blade displacements over time. From the graphs, it is evident that for the phase angles $\Phi = -\pi/2, -\pi/4, 0$, the oscillation amplitude of the middle blade remains limited. This indicates that the oscillation of the blade is damped in these regimes. This situation is well characterized in Fig.11, which shows the energy absorbed by the middle blade from the flow field. A dramatically different situation occurs for the phase angle $\Phi = \pi/4$, where the blade's excitation occurs relatively quickly. This is again evident in Fig.11, where a rapid increase in absorbed energy is shown. This case is highly susceptible to flutter.

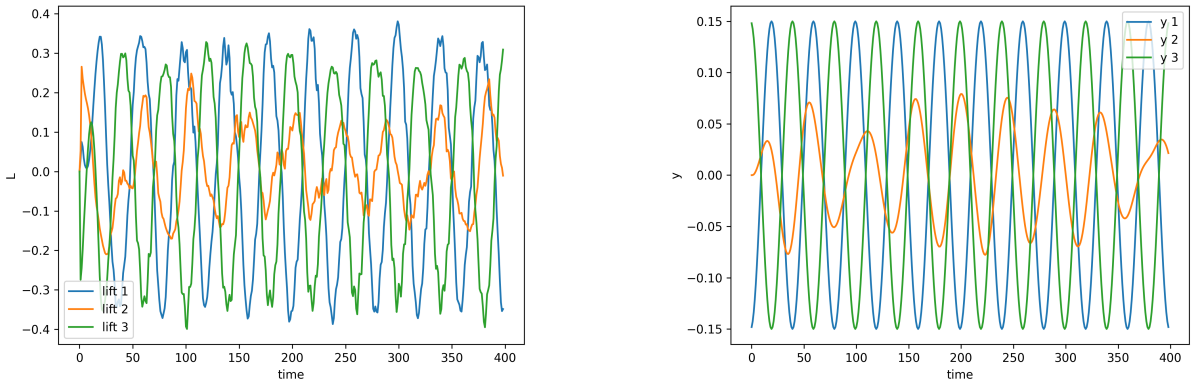

Figure 7: Lift (left) and y-displacement of the middle blade (right) for the phase angle $\Phi = -\pi/2$.

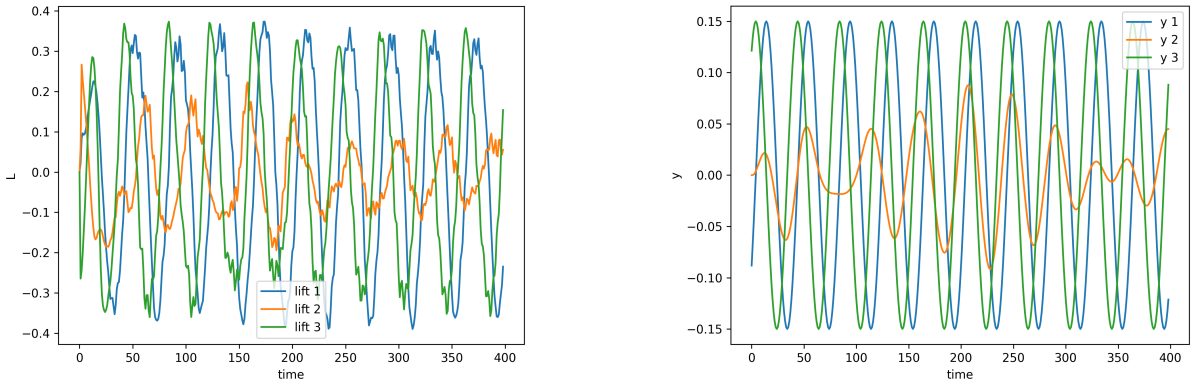

Figure 8: Lift (left) and y-displacement of the middle blade (right) for the phase angle $\Phi = -\pi/4$.

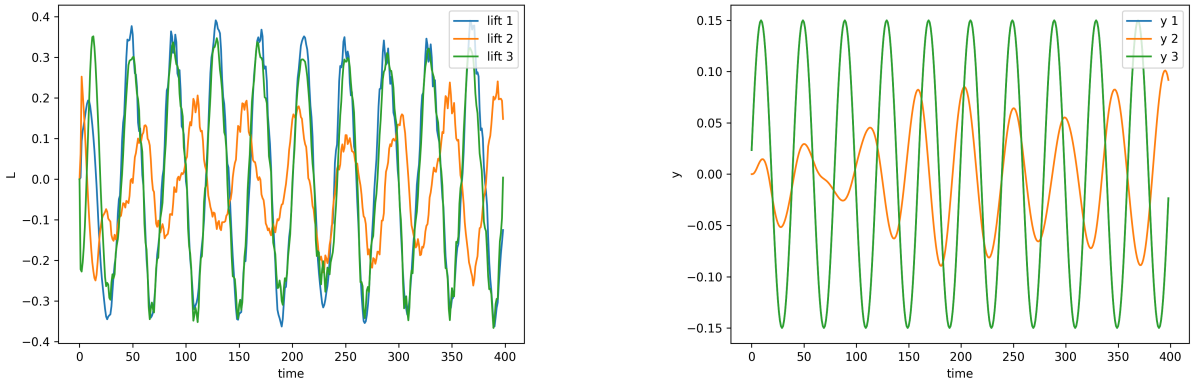

Figure 9: Lift (left) and y-displacement of the middle blade (right) for the phase angle $\Phi = 0$.

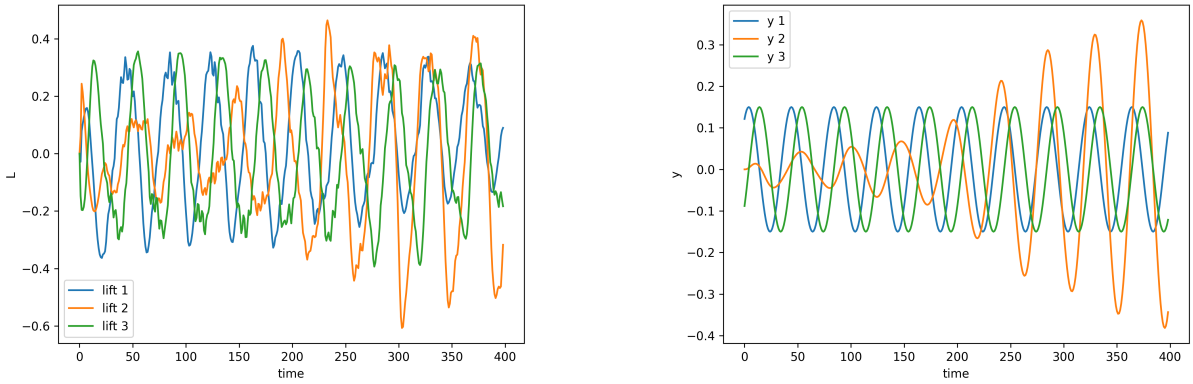

Figure 10: Lift (left) and y-displacement of the middle blade (right) for the phase angle $\Phi = \pi/4$.

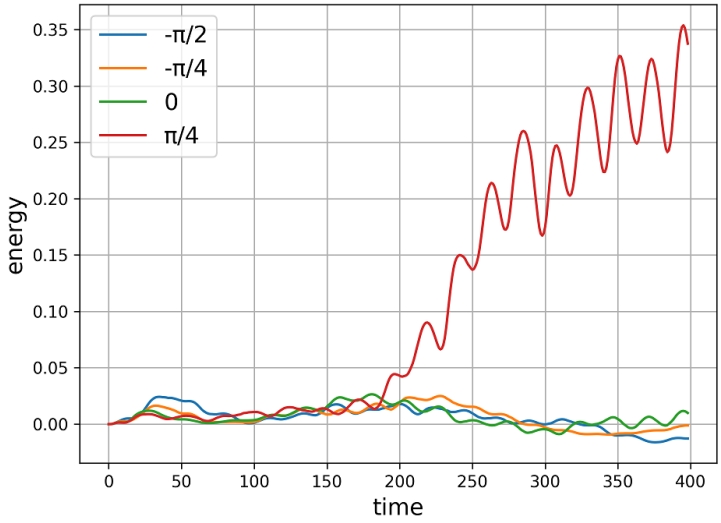

Figure 11: Energy absorbed by middle blade during time.

6 CONCLUSIONS

In this work, a model for predicting unsteady flow in a blade cascade using a convolutional neural network was developed. The model was used to analyze the onset of flutter in a simple cascade consisting of three blades. The analysis showed that for the phase angles $\Phi = -\pi/2, -\pi/4, 0$, there is no excitation of the middle blade. However, for the phase angle $\Phi = \pi/4$, there is an unlimited growth in the oscillation amplitude of the middle blade, leading to flutter. The obtained conclusion is in good agreement with generally accepted findings, where the phase angle $\Phi = \pi/4$ is characteristic of the onset of flutter and corresponds to the so-called travelling waves, which are recorded in turbine disks [7].

The analyzed task demonstrates that the neural network-based model can produce excellent results in a fraction of the computational time. The neural network can generate the flow field up to three orders of magnitude faster than traditional CFD computations. The obtained computational model can be extended to any number of blades using the Schwarz alternating method.

7 ACKNOWLEDGEMENTS

This research is supported by project "Investigation of 3D flow structures and their effects on aeroelastic stability of turbine-blade cascades using experiment and deep learning approach" GACR 24-12144S of the Grant Agency of the Czech Republic.

REFERENCES

[1] Guo, X. and Li, W. and Iorio, F. 2016. "Convolutional neural networks for steady flow approximation." In proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 481-490.

[2] Thuerey, N. and Weißenow, K. and Prantl, L. and Xiangyu, H. 2020. "Deep Learning Methods for Reynolds-Averaged Navier–Stokes Simulations of Airfoil Flows." AIAA Journal

58, no. 1: 25–36.

[3] Bublík, O. and Heidler, V. and Pecka, A. and Vimmr, J. 2022. "Flow-Field Prediction in Periodic Domains Using a Convolution Neural Network with Hypernetwork Parametrization." Int. J. Appl. Mech 15, no. 2.

[4] Bublík, O. and Heidler, V. and Pecka, A. and Vimmr, J. 2023. "Neural-network-based fluid–structure interaction applied to vortex-induced vibration." J. Comput. Appl. Math. 428.

[5] Ronneberger, O. and Fischer, P. and Brox, T. 2015. "U-Net: Convolutional Networks for Biomedical Image Segmentation." arXiv.

[6] Bublík, O. and Vimmr, J. and Jonášová, A. 2015. "Comparison of discontinuous Galerkin time integration schemes for the solution of flow problems with deformable domains." J. Comput. Appl. Math. 267: 329 - 340.

[7] Pust, L. and Pešek, L. 2011. "Vibration of circular bladed disk with imperfections." IJBC 21, no. 10: 2893 - 2904.

[8] Hennigh, O. 2017. "Lat-Net: Compressing Lattice Boltzmann Flow Simulations Using Deep Neural Networks". arXiv:1705.09036.

[9] Lee, S. and You, D. 2019 "Data-driven prediction of unsteady flow over a circular cylinder using deep learning." Journal of Fluid Mechanics, 879:217–254.

[10] Chen, D., Gao, X., Xu, C., Chen, S., Fang, J., Wang, Z., and Wang, Z. 2020 "FlowGAN: A conditional generative adversarial network for flow prediction in various conditions." Proceedings of the 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI), 315–322.