COMPUTATIONAL MECHANICS 2021
36th conference with international participation
Srní
November 8 - 10, 2021

Fast blade shape optimization based on a neural-network-predicted flow field

O. Bublík, A. Pecka, V. Heidler

Faculty of Applied Sciences, University of West Bohemia, Univerzitní 8, 301 00 Plzeň, Czech Republic

The optimization of the geometrical shape of the blade in turbo machinery is generally a computationally demanding task. The optimization algorithms usually use the gradient descent method, which require to compute the flow fields many times to slightly modify the geometry. The use of the neural networks can significantly improve the speed of the optimization process by predicting flow fields extremely quickly [4, 5].

In this work a neural network architecture for prediction of viscous compressible fluid flow in blade cascade was developed. The focus of the architecture is an autoencoder based on the convolutional neural network that transforms a structured computational mesh into the resulting flow field. Periodic boundary conditions were achieved by periodic padding. The developed neural network also contains Mach number as an input parameter. The neural network was implemented using the Python programming language with the help of Keras [3] and TensorFlow [1] libraries.

The developed neural network was trained on 136 randomly generated geometries, where the input Mach number was varied in the range $[0.5, 1]$. The numerical computation of the specimens was performed with open-source CFD software FlowPro [2]. The considered blade profile consist of six design points, where the cubic spline forms the shape of the profile, see Fig. 1. The design points on the tip and the end of the profile are fixed, while the rest of the points can by optimized by finding a maximum of the functional

$$f(\boldsymbol{x}) = \frac{c_L(\boldsymbol{x})}{1 + c_D(\boldsymbol{x})}, \qquad c_L = \oint_\Gamma p\, n_y, \;\; c_D = \oint_\Gamma p\, n_x, \tag{1}$$

where $\boldsymbol{x} = [x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4]$ are positions of the design points, c_L and c_D are lift and drag coefficients and Γ is the profile surface.

The developed neural network was tested on the problem of blade profile optimization for the mach number $M = 1$. In the first step, the optimization algorithm roughly searches the state space for a combinations of the design-point positions, as shown at Fig. 1 (left). The red squares defines the permissible area for the position of free design points. At each of square a nine possible positions were considered (blue points) which lead to a $9^4 = 6561$ combinations. At each of square a nine possible positions were considered (blue points) which lead to a $9^4 = 6561$ combinations. The evaluation of all combination took 13.3s of CPU time on common desktop PC. In the second step, 100 steps of gradient descent method (32s of CPU time) is started to obtain the more precise solution, see Fig. 1 (right). Fig. 2 shows the comparison between predicted and computed pressure and velocity fields for optimized profile.

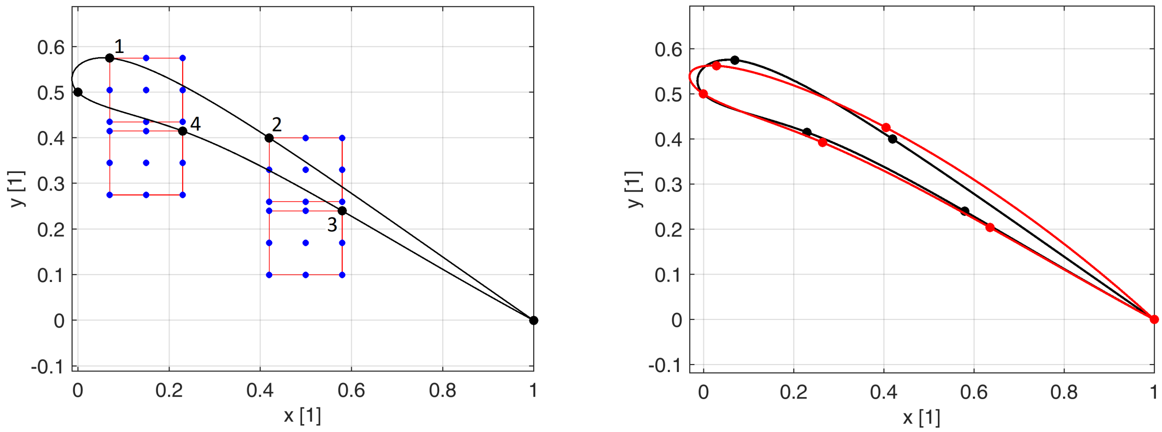

Fig. 1. *Left:* Blade profile with designed points (black dots). Red squares denote the searching area with admissible values (blue points). *Right:* Red color show the optimized profile shape after 100 gradient descent iterations.

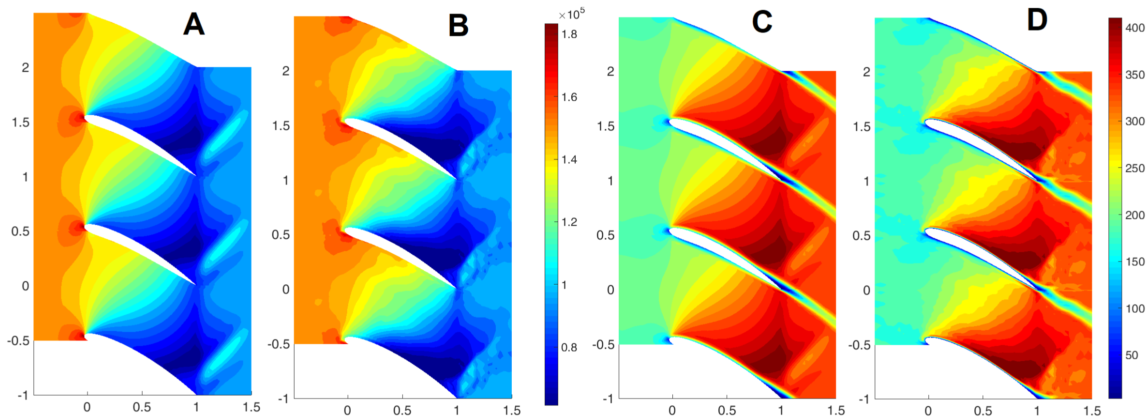

Fig. 2. A: computed pressure field, B: predicted pressure field, C: computed velocity field, D: predicted velocity field

Acknowledgements

This research is supported by project ”Fast flow-field prediction using deep neural networks for solving fluid-structure interaction problems” GA21-31457S of the Grant Agency of the Czech Republic and by the internal student grant project SGS-2019-009.

References

[1] Abadi, M., Agarwal, A., et al., TensorFlow: Large-scale machine learning on heterogeneous systems, Software, 2015, https://www.tensorflow.org/.

[2] Bublík, O., Pecka, A., Vimmr, J., FlowPro – multipurpose CFD software written in Java, Proceedings of the conference Computational mechanics 2017, Srní, 2017, pp. 13-14.

[3] Chollet, F. et al., Keras, 2015, https://keras.io.

[4] Guo, X., Li, W., Iorio, F., Convolutional neural networks for steady flow approximation, Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 481-490.

[5] Hennigh, O., Lat-Net: Compressing lattice Boltzmann flow simulations using deep neural networks, 2017, https://arxiv.org/abs/1705.09036.