

# Parametric model of ventilators simulated in OpenFOAM and Elmer

Václav Čibera<sup>1,a</sup>, Richard Matas<sup>1</sup> and Jan Sedláček<sup>1</sup>

<sup>1</sup>*New Technologies – Research Centre, University of West Bohemia, Univerzitni 8, 306 14 Pilsen, Czech Republic*

**Abstract.** The main goal of presented work was to develop parametric model of a ventilator for CFD and structural analysis. The whole model was designed and scripted in freely available open source programmes in particular in OpenFOAM and Elmer. The main script, which runs or generates other scripts and further control the course of simulation, was written in bash scripting language in Linux environment. Further, the scripts needed for a mesh generation and running of a simulation were prepared using m4 word pre-processor. The use of m4 allowed comfortable set up of the higher amount of scripts. Consequently, the mesh was generated for fluid and solid part of the ventilator within OpenFOAM. Although OpenFOAM offers also a few tools for structural analysis, the mesh of solid parts was transferred into Elmer mesh format with the aim to perform structural analysis in this software. This submitted paper deals namely with part concerning fluid flow through parametrized geometry with different initial conditions. As an example, two simulations were conducted for the same geometric parameters and mesh but for different angular velocity of ventilator rotation.

## 1 Introduction

The work presented in this paper was predominantly focused on a design and simulation of ventilators. The main aim was to develop the tool as universal as possible to design shape and examine the working effectiveness of ventilators at the scale range starting at centimetres scale up to scale of metres. The part of the work included in the text bellow was mainly oriented on the part concerning the properties of fluid flow through the ventilator geometry. Further, the mesh preparation of the solid parts was considered and prepared for structural analysis.

Apart from the main aim of work, i.e. simulation of ventilators, the second main aim was to test and develop the simulation scripts within non-commercial open source software packages. Accordingly, all of the results presented here were achieved with freely available software. In particular, the mesh generation and simulations of fluid flow properties was conducted within OpenFOAM. The mesh of solid part was prepared for structural analysis in Elmer.

Because the ventilators are considered as rotating objects that rotate around their centres, the problem was simplified namely on the account of computing time. In this sense, only repeated segment of ventilator was considered and solved by scripts. Beside this, parallel computation is nowadays a must and matter of course, therefore almost every processes was parallelized. This was achieved namely due to wide range of possibilities in OpenFOAM and Elmer. Last but not least, for a post-processing was used program ParaView, which is also freely available.

<sup>a</sup> Corresponding author: [vacibera@ntc.zcu.cz](mailto:vacibera@ntc.zcu.cz)

The work and the scripts are still under development, but the simulations already work sufficiently for steady – states regardless of the values of input parameters.

The paper is organised as follows. After an introduction section, the methods section follows. In this section, the used software and other tools which were used are introduced. In this part of the text are also presented some effects of the various input parameters. This section contains scripts description as well. But according to the scope of this paper, this description is very simplified. Following section contains results of simulations. The last section is conclusion discussing the summary results.

## 2 Methods

The whole process of solution is a sequence of particulars commands arranged in the main bash script within Linux environment.

### 2.1 Preparation of meshes and script files

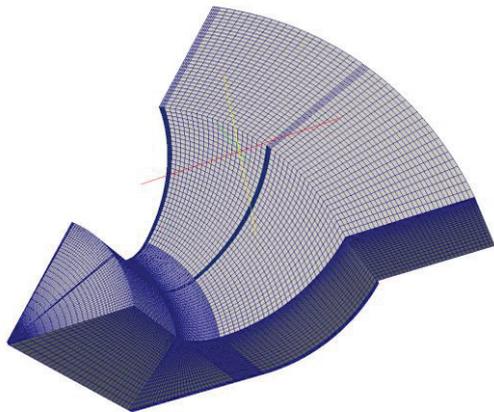
#### 2.1.1 Generation of script files

First of all, it was necessary to spread out input parameters thorough the higher amount of predefined scripts. This was achieved by using m4 word pre-processor. Model inputs within m4 files are then: inlet inner radius, inlet outer radius, inlet height, rotor height, outlet length, blade length, blade width, width of solid part. The meaning of the parameters might be clear from

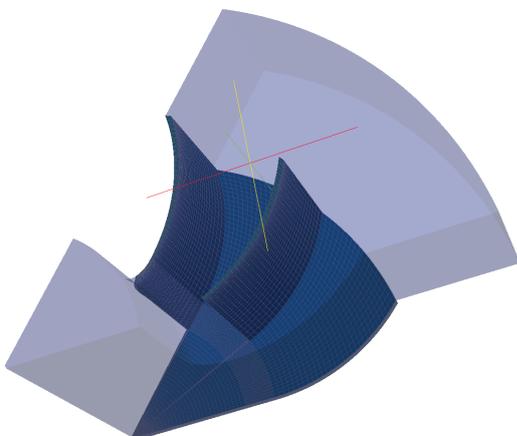
the figures 1 and 2. Additional parameters might be set for the quality of mesh.

The coordinates of the mesh vertices were computed in Octave (also freely available software), when it was called within m4 word pre-processor. The meshes of the fluid part and solid part could be as well generated separately within particular tools of OpenFOAM and Elmer. Separately generated meshes might also avoid the problems with conversion between meshes format from OpenFOAM to Elmer. This might also lead to the more comfortable setup of initial and boundary conditions in Elmer. But since these two meshes were generated within one meshing tool – blockMesh, it satisfies that the nodal points at the common patches (faces) between solid and fluid parts have exactly the same coordinates. This leads to the simplification of transferring for example the data of the pressure on the walls obtained from fluid flow simulation to the structural simulation of the solid part.

Main bash script then contains sequence of commands something like: m4 files.m4, blockMesh -case fluid, createPatch, checkMesh, decomposePar, mpirun -np X simpleFoam - parallel, blockMesh - case solid, createPatch, foamMeshToElmer, ElmerSolver case.sif.

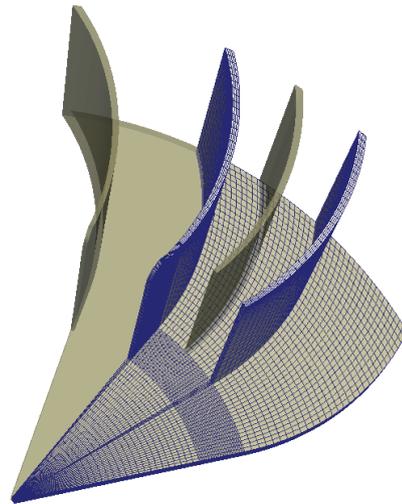


**Figure 1.** Mesh of the ventilator's fluid and solid parts loaded within ParaView. Depicted mesh represents a ventilator consisting of six main blades.

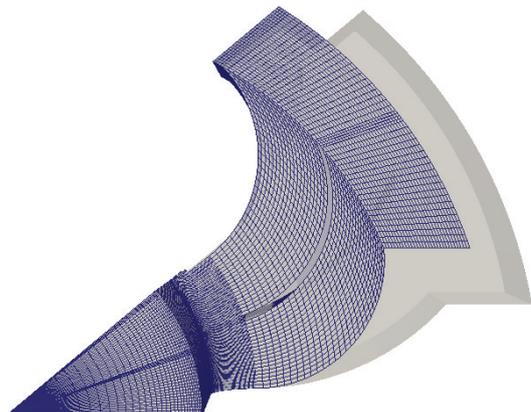


**Figure 2.** Mesh of the ventilator's solid part loaded within ParaView. The picture shows only rotor of solid part incorporated within fluid part.

## 2.2 Examples of the geometric parameters effects



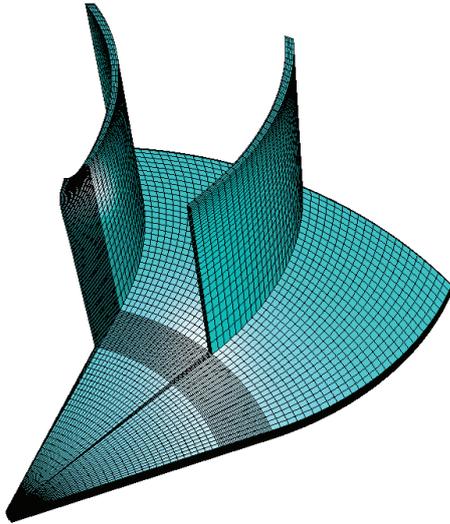
**Figure 3.** Effect of the number of blades in ventilator. The blue part depicts the ventilator with 9 main blades. The grey part depicts the ventilator with 6 main blades.



**Figure 4.** Effect of the blade's radius change. The blades in blue meshed part have radius of 0.03 m and the radius of blades in grey area was set to 0.06 m.

## 2.3 Solution in OpenFOAM

Regarding to the use of the OpenFOAM, two “versions” can be found. This might be confusing. Namely it is OpenFOAM as accessible at [3] and secondly OpenFOAM Extend Project as available at [4]. Both of the versions have the most of the tools in common. But owing to the fact, that each version is developed by different groups, some required tools were missing either in first or the second version. Therefore at the end, both of the versions were used. The first noted version was used primarily to prepare meshes and for simulation of fluid parts of ventilator. The later noted version was predominantly used for preparation of the solid part of mesh. Specifically it was used for the conversion of the mesh from OpenFOAM format to Elmer mesh format.



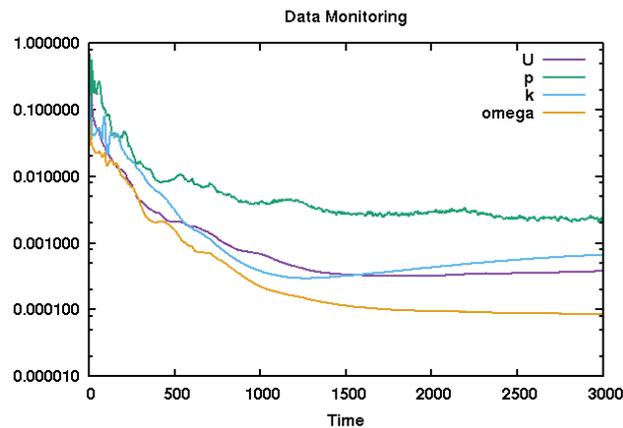
**Figure 5.** Mesh of the ventilator's solid part loaded within ElmerGUI after transferred from OpenFOAM mesh format to Elmer mesh format.

### 3 Results

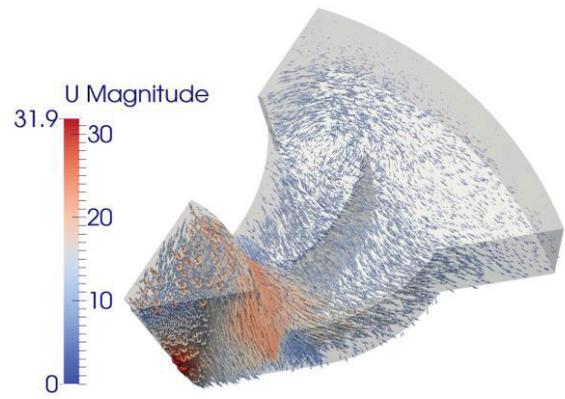
As an example, two simulations were conducted. Both simulations were computed as the steady-state simulations. In both cases, the same geometric parameters of ventilator were used. Therefore the same mesh was generated. The best results reflecting reality were achieved with k-omega SST turbulence model. The fluid medium was considered as incompressible air. The boundary conditions for pressure were set as absolute atmospheric pressure at the inlet and outlet of ventilator. Consequently, flow velocity field and pressure field within ventilator geometry was computed.

#### 3.1 Results for simulation for 1000 and 2000 rpm

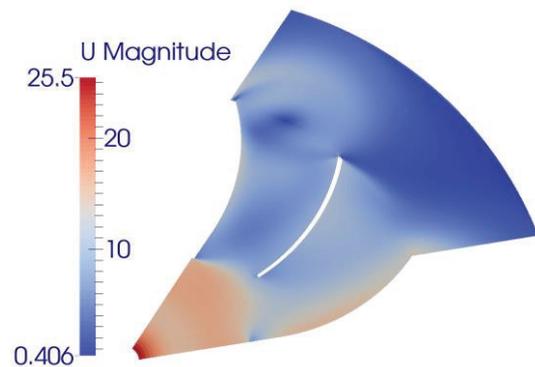
The results as presented here were conducted for 1000 and 2000 rpm of ventilator. The tasks of steady-state solution converged within 3000 iteration to the acceptable values of residuals as well as for various rpm.



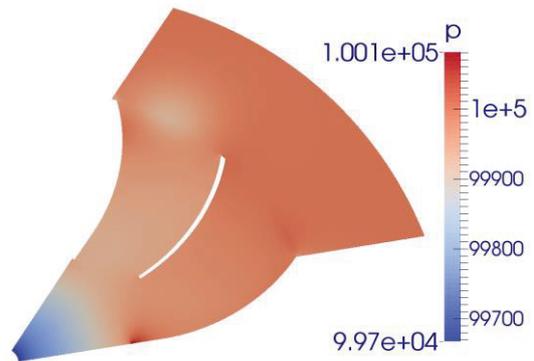
**Figure 6.** Convergence of simulation residuals. The picture is output from tool foamMonitor. The x-label Time means here in a fact a number of iteration (steady-state solution).



**Figure 7.** Velocity field obtained for the simulation of the 1000 rpm.



**Figure 8.** Velocity fields in the centre plane of the rotor achieved by 1000 rpm.

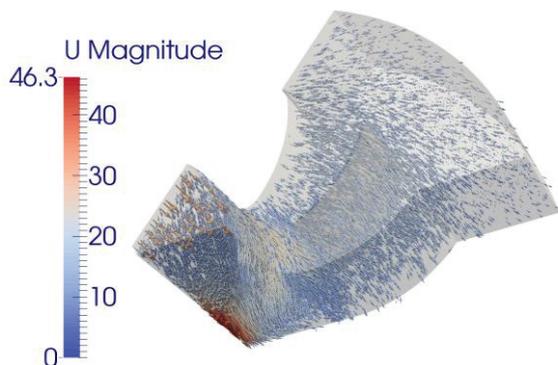


**Figure 9.** Absolute pressure in the centre plane of the rotor achieved by 1000 rpm.

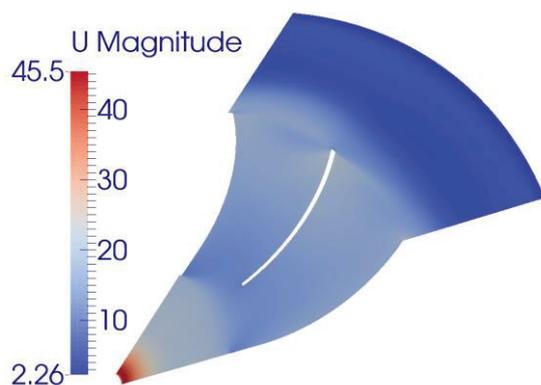
The range of the velocity magnitudes as depicted in figure 7 does not represent the maximal value in the ventilator volume since the velocity vectors were generated randomly. Therefore the figure 7 is predominantly for illustration of the direction of the velocity fields. Hence this picture can be hard compared with the same kind of figure 10 for 2000 rpm in absolute values.

The values that can be compared on account of the change of the angular velocity of ventilator are depicted in the figures 8, 9 and 11, 12. In the figures 8 and 9 are shown velocity fields in the cross-section through the rotor. The cross-section is situated exactly in the middle of rotor height. As can be seen in these

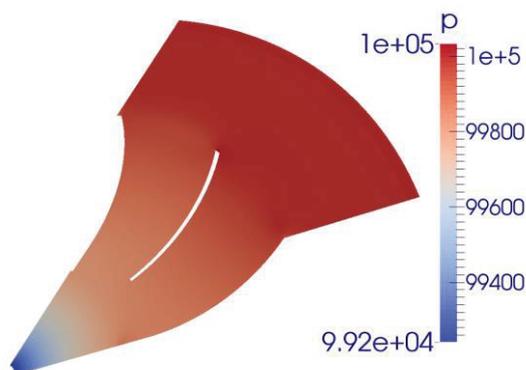
pictures, for the higher rpm of ventilator, the higher velocities were achieved which is theoretically in concordance with reality. But for the more accurate characteristic of ventilator, more states should be computed. Consequently, pressure fields were not changed significantly since the simulation supposes incompressible flow with atmospheric pressure at the outlet as well as inlet of ventilator.



**Figure 10.** Velocity field obtained for the simulation of the 2000 rpm.



**Figure 11.** Velocity fields in the centre plane of the rotor achieved by 2000 rpm.



**Figure 12.** Absolute pressure in the centre plane of the rotor achieved by 2000 rpm.

## 4 Conclusions

Before all, it is worthy to notice that in comparison to commercial software, the pre-processing in open source programs as in OpenFOAM or Elmer might take longer time from various reason such as lack of documentation or lack of graphical user interfaces. This might be disadvantage namely in the cases, were it is demanded only a small number of simulations for small number of parameters variations.

The advantage of the use of the mentioned might be found in the cases as the solution introduced here. The reason for this is that since this software is open-source, new tool can be comfortably added, which facilitates scripting of the more extensive project. This way the solved project can be more comfortably prepared on account of the specific requests.

The results were compared with results achieved by using ANSYS FLUENT 15. The version that were used for OpenFOAM were 2.4, 2.3 respectively for OpenFOAM Extended Project. The same mesh, turbulent models etc. were used in both CFD solvers. The only parameters that were not set equally were the numerical solvers. Comparing the results of maximal velocities, the differences in absolute values were no higher than 3%.

Simulations usually converged to required accuracy within 3000 iteration. The results were achieved in the range of minutes for small meshes and in the range of tens of minutes.

According to the generation of the scripts by m4 word pre-processor, a little bit surprising was the slowness of generation. It was believed that it was caused by calling of Octave to compute the parameters of the mesh.

By using developed scripts, optimal working regime and characteristics properties of the ventilator with particular geometry can be found. This way, the various ventilators can be tune up for an ideal angular velocity, working regimes and manufacturing process.

The scripts are still being developed. The next step in the development is to include scripts, which would automatically compute modal analysis of solid parts. Last but not least, further the codes could be optimized for transient solution.

## Acknowledgement

The work was supported by internal grant at University of West Bohemia: SGS-2014-070: "Zvyšování účinnosti, spolehlivosti a životnosti energetických strojů a zařízení 3".

## References

1. J.H. Ferziger, M. Perić, Computational Methods for Fluid Dynamics, third, rev. edition. Springer 2002.
2. C. Hirsch, Numerical computation of internal and external flows, volume 1, John Wiley and Sons, 1997
3. [www.openfoam.com](http://www.openfoam.com)
4. <http://www.extend-project.de/>