

Scalability Issues for Wind Simulation Using OpenFOAM

*Neki Frasheri*¹, *Emanouil Atanassov*²

¹*Faculty of Information Technology, Polytechnic University of Tirana, Tirana, Albania*

²*Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, 1113 Sofia, Bulgaria*

E-mails: nfrasheri@fti.edu.al emanouil@parallel.bas.bg

Abstract: *The use of the OpenFOAM software for wind simulation over rugged terrain is studied in the paper. OpenFOAM requirements for runtime, virtual memory and disk space are considered using small and medium resolution Digital Elevation Models (DEM) models for single-process and multi-process cases when running on local workstation and small parallel systems. Based on obtained data, extrapolation of requirements for relatively high resolution models are made and issues concerning the scalability are discussed.*

Keywords: *OpenFOAM, wind simulation, rugged terrain, High Performance Computing (HPC) scalability.*

1. Introduction

In this paper we consider the scalability issues that arise when performing wind simulation using the popular open-source software OpenFOAM. The H2020 project VI-SEEM aims at creating a unique Virtual Research Environment (VRE) in Southeast Europe and the Eastern Mediterranean (SEEM), with special focus on the scientific communities of Life Sciences, Climatology and Digital Cultural Heritage [<https://vi-seem.eu/>]. In this framework one of tasks is to realize wind simulation over rugged terrain, taking as model the mountainous area that includes Albania. This task is mainly performed by the Polytechnic University of Tirana (UPT). The terrain model was obtained from NASA Shuttle Radar Topography Mission (SRTM) Digital Elevation Model [USGS archive: <https://lta.cr.usgs.gov/>]. Experiments were carried out running the software on local workstations and multiprocessor systems, aiming at the evaluation of the runtime requirements as a preparatory phase for running it in the VI-SEEM VRE. A preliminary literature review showed that the main scalability problems to be expected depended on the problem to be solved, the model size, the inter-process communications, etc., [1-13]. In most of the reviewed literature the analysis of scalability was done for

concrete problems, without giving details on memory usage, which we found to be critical in our experiments, confirming the remark of Culpo that “the size of the problems that can be handled on a High Performance Computing (HPC) cluster lies beyond the limitations imposed by smaller in-house clusters” [6]. The paper is organized as follows: we first describe the use case that we consider, then we describe some preliminary results and then we present and discuss the main scalability results and finish with conclusions.

2. Use case setup

Mountainous ranges in Western Balkans are characterized by narrow deep valleys that require high-resolution models for wind simulation. Free SRTM DEM that covers Albania is with a resolution of 3 arcsec per 1 pixel, which corresponds to a rectangular resolution of 100×100m on the equator. We selected a section of DEM of higher resolution available with size 3600×4800 pixels (Fig. 1).

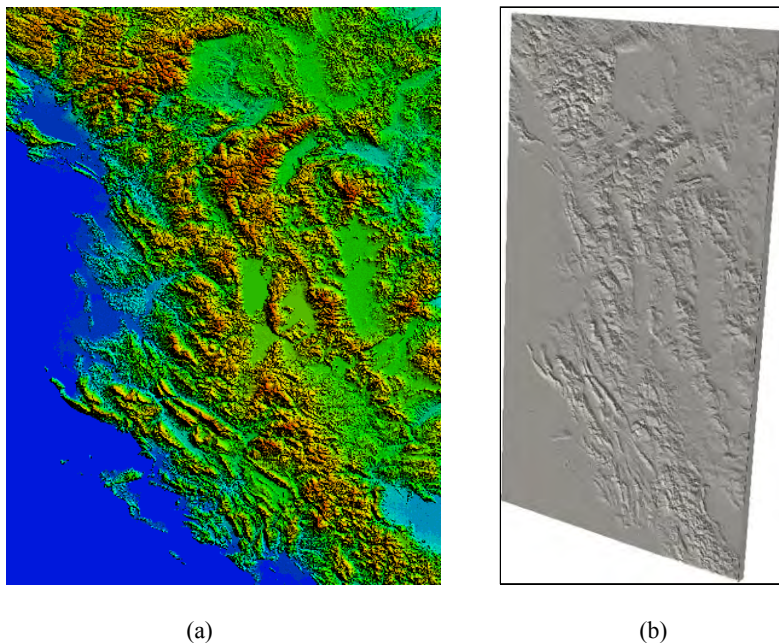


Fig. 1. DEM image for Albania and surrounding area (a); bottom surface of 3D volume deformed based on DEM (b)

The 3D model size in kilometres was 270×480×10 km taking into account the variation of arcsec distances by latitude. In order to test the scalability two other principal models of medium size 360×480 pixels and low size, 36×48 pixels, were considered. During the experiments, models with intermediate sizes were used as well. The results are presented in Table 1, where one can see the number of digitized elements for each model and the requested virtual memory RAM.

Table 1. Model sizes used for experiments

Factor	3D Volume Size			Data Size	
	X-size	Y-size	Z-size	Elements	RAM, KB
10	36	48	10	1.73×10^4	1.20×10^5
20	108	144	30	4.67×10^5	4.98×10^5
43	154	206	43	1.36×10^6	1.25×10^6
60	216	288	60	3.73×10^6	3.07×10^6
100	360	480	100	1.73×10^7	1.31×10^7
139	500	667	139	4.64×10^7	3.34×10^7
1000	3600	4800	1000	1.73×10^{10}	1.43×10^9

The preparation of geographical data for use in OpenFOAM represents a separate problem [15, 16]. We transformed our SRTM DEM data into Surfer ASCII array format with the aid of software GDAL [http://www.gdal.org/frmt_various.html]. A small program was developed to modify the grid, generated by OpenFOAM module blockMesh, to reflect DEM altitudes for the shaping of the relief in the bottom frame of the grid.

Comprehensive experiments were carried out running the OpenFOAM solver for 100 time steps of length 0.01 s, storing results for only 10 times. Time steps of 0.1 and 1.0 s lead to divergence of the process for higher resolution models due to an increase of Courant number bigger than 1.0 as result of dis-balancing between spatial and temporal discretization steps [14].

The computer systems used in UPT included a single computer with Intel core I7 and 16 GB of RAM, and a small parallel system (SUGON) based on Intel Xeon E5506 processors but with only 2GB RAM per node; the operating system was Linux Fedora 20 on the desktop and Scientific Linux 6.7 on the SUGON.

Further experiments were performed on an High Performance Computing (HPC) Cluster with non-blocking DDR Infiniband with 36 nodes with dual Intel(R) Xeon(R) CPU X5560 @ 2.80GHz and 24 GB RAM; and the supercomputer system AVITOHOL with 150 servers with dual Intel Xeon E5-2650v2 8C @2.6GHz; both at the Institute of Information and Communication Technologies – Bulgarian Academy of Sciences (IICT-BAS).

3. Preliminary simulation results

SRTM DEM data are presented in Fig. 1. The area of study includes Albania, and parts of Montenegro, Kosovo, Macedonia and Greece.

The area is characterized by a group of high mountains in North Alps, and several ranges of mountains that extend in direction North-North-West: South-South-East. Only about 1/3 of the territory of Albania is lowlands, the so-called PreAdriatic Depression. Narrow valleys have their impact in wind flows.

The 3D prismatic layer area of thickness 10 km was digitized which resulted as the output of blockMesh module was deformed modifying Z-coordinates of all nodes, obtaining the shape of the terrain in the bottom face of the prism as in Fig. 1.

Some examples from execution of the model with linear size factor 100 are presented hereunder. Simulation was done for boundary conditions with fixed potentials of value 0 and 1 in southern and northern faces respectively that give a wind flow North-to-South.

Simulation images are obtained using the ParaView software for the velocity U in altitude of 1000 m. The North-South velocity of wind flow in altitude 1000 m is presented in Fig. 2a. Gray areas represent mountain altitudes over 1000 m.

A near-to-surface slowness of the wind is visible around mountain peaks. The effect of the relief is more visible in the following images representing West-East and vertical wind-flows.

Due to oblique extension of mountain ranges, in all valleys there is generated a component of West-East wind flow with magnitude 50% in the altitude 1000 m (Fig. 2b). At the same time “protected” valleys where this wind has only a magnitude of 25% are identified.

The vertical flow of the wind in altitude 1000m reaches high velocities in some of the mountain slopes (Fig. 2c). This phenomenon is significant in two areas – northern boundaries of Mirdita and mountainous areas East of Vlora city. Such results are important for planning of setup of wind energy farms, and for air transport. A more complete analysis is planned for the future work.

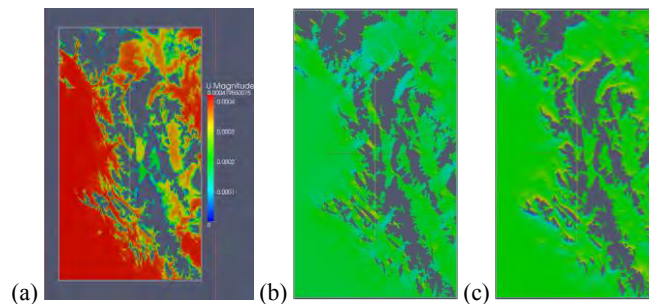


Fig. 2. North-South wind flow in altitude 1000 m (a); induced West-East wind flow in altitude 1000 m (b); induced vertical wind flow in altitude 1000 m (c)

4. OpenFOAM scalability results

The execution of OpenFOAM suite in single-process mode requires the run of the mesh generator blockMesh, followed in our case by the solver icoFoam. After the solver the other modules like ParaView may be used for the visualization of results. While the mesh generator and the solver may be run remotely through Secure Shell (SSH) sessions, for a remote visualization of results it is necessary to use ParaView in client-server mode. In case of MPI parallel run of the solver two other OpenFOAM modules are needed for the decomposing of the mesh in fragments for each process, and the combination of fragmented results in a single set. Decomposing can be arranged in different ways, and we tried several decomposition schemes with 16 fragments for 16 parallel MPI processes, with runtime presented in Table 2.

Table 2. Model runtime for decomposition schemes of 3D area

Decomposing schema	Runtime for software packages, s		
	decompose	icoFoam	reconstruct
(4:4:1)	3.9098×10^2	2.6090×10^3	1.7554×10^3
(4:1:4)	6.7977×10^2	4.1040×10^3	3.2037×10^3
(1:4:4)	3.9362×10^2	4.1690×10^3	3.3695×10^3
(16:1:1)	6.7534×10^2	2.7570×10^3	3.1734×10^3
(1:16:1)	6.8242×10^2	2.5450×10^3	3.2331×10^3
(1:1:16)	7.6215×10^2	8.2140×10^3	3.5776×10^3

Detailed analysis of scalability was done for the first case of decomposition of X and Y axes (4:4:1). The runtime was evaluated using the log of the icoFoam solver and by running the software through the /usr/bin/time launcher.

The initial data for the OpenFOAM are not voluminous; they include text files for different physical and temporal parameters, coordinates of basic blocks of the area and respective faces. The module blockMesh generates huge text files with coordinates of each node and nodes for each face. This data is used as input for the solver that generates, for different time slots in respective directories, text files with values of the potential p per node, also the velocity U and flux ϕ for each face.

The decomposing module generates text files with coordinates of nodes and nodes of faces for each MPI process, and the solver processes generate their results in separate directories. After the reconstruction, there is significant redundancy of data in the disk. Also, the evaluation of required disk space is dependent of the number of time slots recorded. This scheme of exploitation of OpenFOAM as described above was considered during the experiments, through evaluation of both memory, disk and CPU requirements of different modules. When the memory requirements exceeded the available RAM, the machines had to use swap space, which showed bad performance and should be avoided.

We succeeded to run two principal models of low and medium resolution on the local computer, obtaining first evaluations for virtual memory requirements and disk data volumes, presented in Fig. 3. The order of virtual memory requirements resulted at the range of $O(N^2)$, while for the disk space they were $O(N^3)$, where N is the number of nodes in one linear edge of the 3D digitized area. The medium-sized model of 360×480 points was run on the local computer with 16 GB RAM, but could not be executed on the local parallel system of UPT due to the small amount of available RAM. Running the high resolution model simulation as a single process for the DEM with 3600×4800 points (the factor 1000) would require a system with RAM of the range of 2000 GB; smaller amount of memory would increase significantly the runtime due to disk swapping. In all cases the values for the high-resolution model were extrapolated based on low and medium model results (the extrapolated values are colored in red).

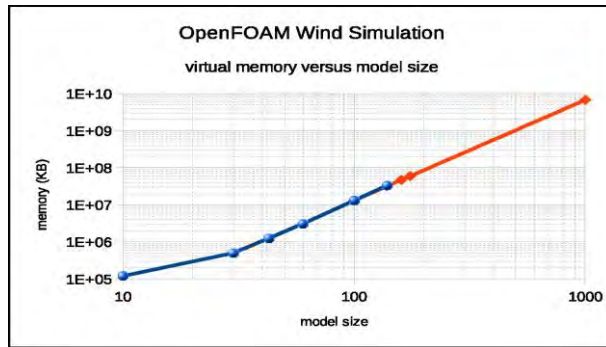


Fig. 3. Memory usage of OpenFOAM – single process run

In the case of multi-process execution of the OpenFOAM solver we obtained another view of the memory requirements presented in Fig. 4. An asymptotic tendency is observed towards the level of 1 GB of virtual memory. The external storage requirements are presented in Fig. 5.

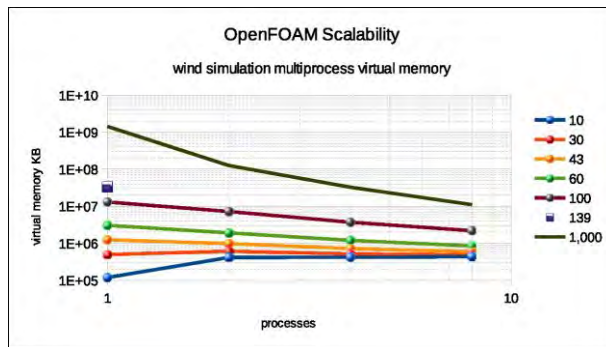


Fig. 4. Memory usage of OpenFOAM – multi process run

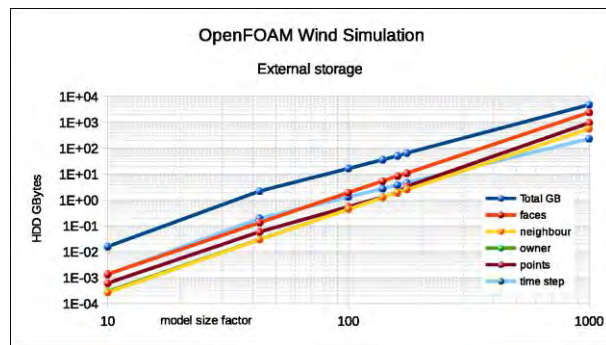


Fig. 5. Disk usage of OpenFOAM versus model size

We considered single process runtime for the OpenFOAM modules – blockMesh, decompose and reconstruct for digitizing of the area, decomposing the data for multiprocess MPI run and recomposing results respectively; in our version of OpenFOAM these modules are not parallelized.

Only the solver module icoFoam is parallelized with MPI and was run in parallel. This fact makes it difficult to run OpenFOAM on small systems even if MPI processes would require less RAM: even if we would be splitting each MPI process in a separate node to use the full amount of RAM, the preparatory tasks must run in a single node. The parallel MPI runtime was evaluated, firstly on the HPC parallel system, and finally on the AVITOHOL system of IICT-BAS in Sofia. The size of RAM in AVITOHOL permitted the execution of medium sized models without memory swapping, giving cleaner results for the runtime presented hereunder.

The runtime versus model sizes in AVITOHOL is presented in Fig. 6, while versus number of MPI processes it is presented in Fig. 7.

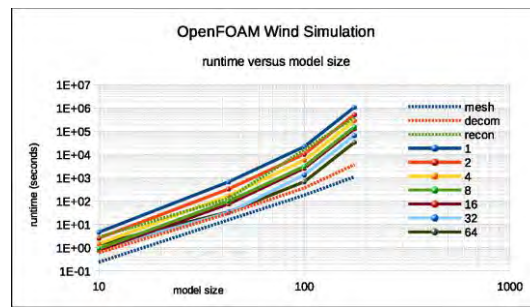


Fig. 6. OpenFOAM runtime versus model size factor

The runtime for the low-resolution model with more than 16 processes degenerates due to volume of inter-process communications in rapport with the volume of data, the same observations are obtained for a higher resolution (factor of 43) when more than 32 processes are used.

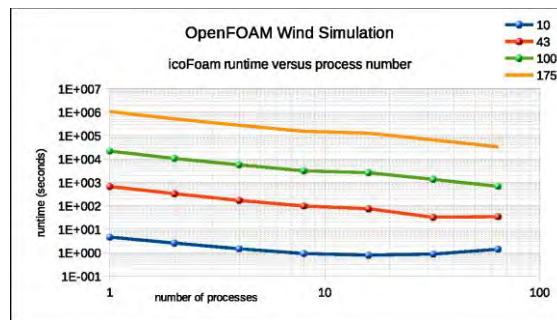


Fig. 7. OpenFOAM runtime versus MPI process number

For higher-resolution models there is a slight increase of runtime when more than 16 processes are used – Avitohol nodes have only physical 16 cores each. The analysis of solution errors was based in initial residuals for parameters P , U_x , U_y , U_z for each time iteration, and related courant numbers. Using longer than 0.01 s time step, solution process degenerated for medium resolution models, presented in Fig. 8. For convergent solutions with time step 0.01 s the variation of residuals and related courant numbers is presented in Fig. 9.

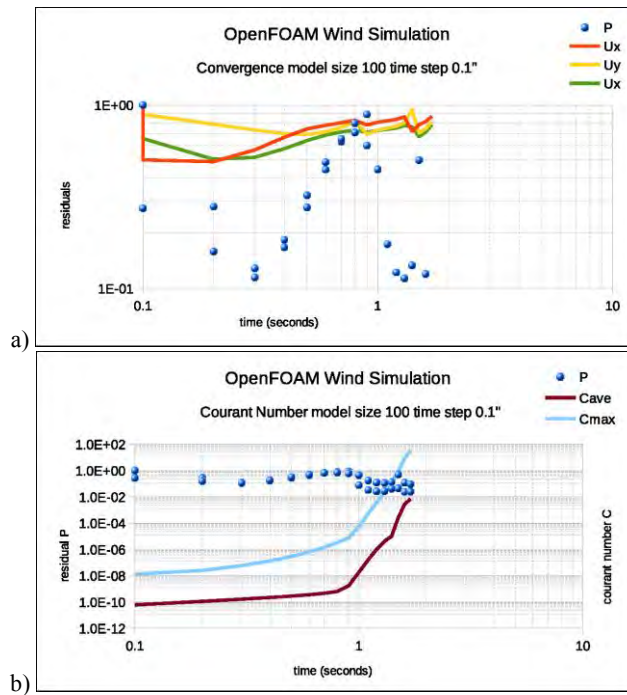


Fig. 8. Divergence of solution processes for time steps 0.1 s: Initial residuals (a); average and maximal courant numbers (b)

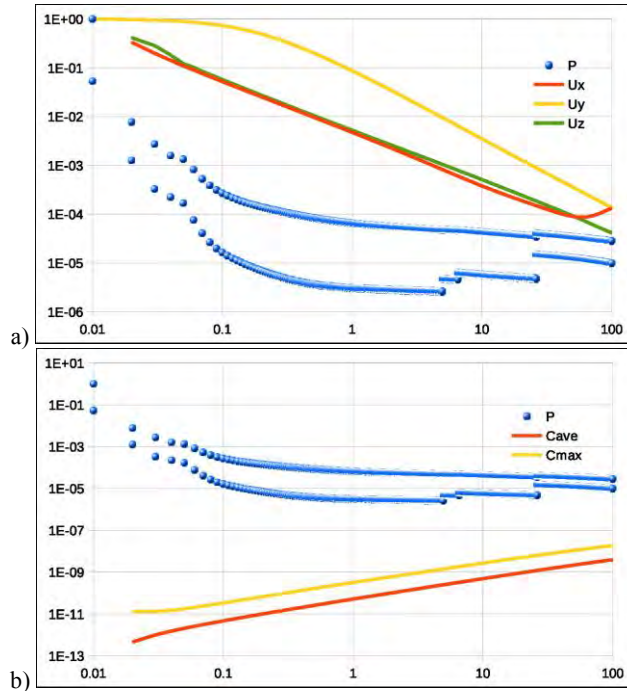


Fig. 9. Convergence of solution process for 10,000 time iterations: Initial residuals(a); average and maximal courant numbers (b)

5. Conclusions

The Usage of OpenFOAM for wind simulation over rugged mountainous terrain, for the territory of Albania and surrounding area, resulted as problematic in small parallel systems. The amount of requested memory reached levels of 10-30 GB for medium-sized models, which resulted to be higher than the available RAM causing degradation of the runtime and the scalability. Even when the RAM is sufficient, the increase of the number of processes leads to degradation of scalability. Actually the decreasing multi-process runtime for medium-sized models was obtained only in high-range HPC systems like Avitohol.

Considering the high-resolution model, an extrapolation of runtime gave the value on the range of 10,000,000 s (3000 h) with 2000 GB total virtual memory. Only supercomputer systems like Avitohol may offer such resources (it has 9600 GB). It is also necessary to consider that for the study of turbulence and eddies, it would require the storage on disk of a greater number of time slots, compared with experiment setup presented in this paper.

The parallel execution of OpenFOAM solver may require much less virtual memory for higher number of processes but auxiliary calculations for mesh generation, decomposing the data for each process and recomposing results apparently cannot be run in parallel, requiring the full central memory capacity.

Acknowledgments: This work was supported by the European Commission under H2020 Project VI-SEEM (Contract Number 675121).

References

1. Lombardi, M., N. Parolini, A. Quarteroni, G. Rozza. Numerical Simulation of Sailing Boats: Dynamics, FSI, and Shape Optimization. MATHICSE Technical Report No 03.2011, April 2011.
2. Ravelli, S., G. Barigozzi, F. Pasqua, R. Pieri, R. Ponzini. Numerical and Experimental Study for the Prediction of the Steady, Three Dimensional Flow in a Turbine Nozzle Vane Cascade Using OpenFOAM. – In: Proc. of International CAE Conference 2014, Verona, Italy, October 2014, pp. 27-28.
3. Lysenko, D. A., I. S. Ertesvag, K. E. Rian. Towards Simulation of Far-Field Aerodynamic Sound from a Circular Cylinder Using OpenFOAM. – Aeroacoustics, Vol. **13**, 2014, No 1.
4. Karasek, T., D. Horak, V. Hapla, A. Markopoulos, L. Riha, V. Vondrak, T. Brzobohaty. Application of CFD and CSM Open Source Codes for Solving Multiscale Multiphysics Problems. IT4 Innovations, VSB, Technical University of Ostrava. www.prace-ri.eu
5. Abdi, D. S., G. T. Bitsuamlak. Asynchronous Parallelization of a CFD Solver. – Hindawi Publishing Corporation Journal of Computational Engineering, Vol. **2015**, 2015, Article ID 295393. <http://dx.doi.org/10.1155/2015/295393>
6. Culp, M. Current Bottlenecks in the Scalability of OpenFOAM on Massively Parallel Clusters. Partnership for Advanced Computing in Europe, 2010. <http://www.prace-ri.eu/IMG/pdf>

7. Lysenko, D., I. Ertesvag, K. Rian. Testing of Open-FOAM CFD Code for Plane Turbulent Bluff Body Flows within Conventional URANS Approach. – In: Proc. of International Conference on Computational Fluid Dynamics in the Oil Gas, Metallurgical and Process Industries (CFD'11), Trondheim, Norway, June 2011.
8. Kornyei, L. Simulation of Gas Flow in a Combustion Chamber Using High Performance Computing Hardware. – In: Proc. of Workshop on the Occasion of the 60th Birthday of Ferenc Igloi, Budapest, Hungary, 3 October 2012.
9. Tanasescu, C. OpenFOAM and SGI Designed to Work Together.
<http://docplayer.net/3118206-Openfoam-and-sgi-designed-to-work-together-christian-tanasescu-vice-president-software-engineering.html>
10. Dor, A. L., G. M. Laskowski, A. Gupta. Investigating Primary Breakup with OpenFoam: Performance and Validation. – In: Proc. of International Conference for High Performance Computing, Networking, Storage and Analysis (SC'10), New Orleans, USA, November 2010, pp. 13-19.
11. Ponweiser, T., P. Stadelmeyer, T. Karasek. Fluid-Structure Simulations with OpenFOAM for Aircraft Designs. Partnership for Advanced Computing in Europe (PRACE) Report.
www.prace-ri.eu/IMG/pdf/wp172.pdf
12. Rivera, O., K. Furlinger, D. Kranzlmuller. Investigating the Scalability of OpenFOAM for the Solution of Transport Equations and Large Eddy Simulations. – In: Proc. of 11th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP'11), Vol. **Part II**, LNCS 7017, 2011, pp. 121-130.
13. Sidlof, P., V. Ridky. Scalability of the Parallel CFD Simulations of Flow Past a Fluttering Airfoil in Open-FOAM. – In: EPJ Web of Conferences, Vol. **92**, 2015. DOI: 10.1051/epjconf/20159202080.
14. ESI-OpenCFD. OpenFOAM Lid-Driven Cavity Flow.
<http://www.openfoam.com/documentation/tutorial-guide/tutorialse1.php>
15. Hardin, E. How to Build an OpenFOAM Case from GRASS GIS Digital Elevation Models. North Carolina State University, 2013.
http://www4.ncsu.edu/~ejhardi2/OF_GRASS_Geom.pdf
16. Tapia, X. P. Modelling of Wind Flow over Complex Terrain Using OpenFoam. University of Gävle, Sweden, 2009.
<http://www.diva-portal.org/smash/get/diva2%3A228936/FULLTEXT01.pdf>