# User's Guide for the Very Large-Eddy Simulation (VLES v1.0.0) Software

Mojtaba Ahmadi-Baloutaki and Amir A. Aliabadi

http://www.aaa-scientists.com

aliabadi@uoguelph.ca

This document is typeset using LaTeX

February 2, 2023

The Very Large-Eddy Simulation (VLES) software is developed for OpenFOAM 4.0 on a Linux platform (e.g. UBUNTU). It enables simulations of thermo-fluid systems by solving the transport equations of continuity, momentum, heat, passive scalar, and turbulence kinetic energy. More details on the model can be found at the Atmospheric Innovations Research (AIR) laboratory website at www.aaa-scientists.com and corresponding publications [Aliabadi et al., 2017, Aliabadi et al., 2018, Ahmadi-Baloutaki and Aliabadi, 2021, Kia et al., 2021]. The software release and the following guidelines specifically relate to [Ahmadi-Baloutaki and Aliabadi, 2021].



To install the new solver:

1. Open a terminal window and go to the provided "solvers" directory.
2. Run this command for admin access: `sudo su`.
3. Run this command to clean the previous directories: `wclean`.
4. Run this command to implement the new solver: `wmake`.
5. Be sure that you have not received any error.

To install the new utility:

1. Open a terminal window and go to the provided "utilities" directory.
2. Run this command for admin access: `sudo su`.
3. Run this command to clean the previous directories: `wclean`.
4. Run this command to implement the new utilities: `wmake`.
5. Be sure that you have not received any error.

To install the new library of the developed Synthetic Eddy Method (SEM):

1. Open a terminal window and go to the provided "derived" directory.
2. Run this command for admin access: `sudo su`.
3. Run this command to clean the previous directories: `wclean`.
4. Run this command to implement the new SEM libraries: `wmake libso`.
5. Be sure that you have not received any error.

There are four cases related to wind tunnels that can be used as follows:

1. Copy the cases to the "run" folder of the OpenFOAM user.
2. "CaseUwithoutWallFunction" solves the momentum equation for a tunnel case without using any wall function for regions near the wall.
3. "CaseUwithWallFunction" solves the momentum equation for a tunnel case with using a wall function for regions near the wall. The momentum wall function is called "nutkAtmRoughWallFunction" and was implemented in the "nut" file in the "0" folder of the case.
4. "CaseUTwithoutWallFunctions" solves the momentum and energy equations for a tunnel case without using any wall function for regions near the wall.
5. "CaseUTwithWallFunctions" solves the momentum and energy equations for a tunnel case with using wall functions for regions near the wall. The energy wall function is called "alphatJayatillekeWallFunction" and was implemented in the "alphat" file in the "0" folder of the case.
6. Just as a note, "CaseUT"s generate local libraries in a folder called "dynamicCode". This is for implementing a "codedFixedValue" boundary condition for temperature, where we define a fixed power law for inlet temperature.

The main procedure of setting up a case in OpenFOAM is as follows:

1. Boundary conditions are defined in the folder "0" of the case. They should be modified based on each new run if it is necessary. In the provided cases, required parameters of the

boundary conditions are set in the file "ABLConditions", except for constants that define the temperature boundary conditions. These have to be inserted under the "0/T" for the "codedFixedValue" type: Tinf, Ts, Zmax, and alpha.

2. Physical properties of the case, such as transport and turbulence properties, are defined in the folder "constant". They should be modified based on each new run if it is necessary.

3. Mesh settings are defined in the folder "constant", the subfolder "polyMesh", and the file "blockMeshDict". They should be modified based on each new run if it is necessary.

4. Numerical schemes and solution parameters are defined in the folder "system". They should be modified based on each new run if it is necessary.

5. If running in parallel is required, number of cores in the file "decomposeParDict" of the folder "system" should be modified based on the available computational cores in each computer; keeping two cores aside is recommended for other common activities.

6. In the file "controlDict" of the folder "system" run controls should be modified based on each new run. In addition of the usual settings, in the provided cases, "probing" is also available. As a rule of thumb, comment the function part of the mentioned file (by adding "/*") for the first 12 seconds because of reaching to the stable solution. This corresponds to two complete passes of the flow over the entire stream-wise direction. (Note: some early simulations only let 5 seconds before probing the solution, but 12 seconds is more conservative). After that, uncomment the "functions" part to save the solution variables for each probe specified in a folder called "postProcessing" usually for more 40 seconds of run. Right now there are 6 profiles with about 200 probes on each.

For running a case in series, simply follow this procedure:

1. Open a terminal window and go to the case directory.

2. Run: `passiveScalarLESBuoyantBoussinesqPimpleFoam`.

3. For calculating the vorticity and having it for post-processing part, run: `postprocess -func vorticity`.

For running a case in parallel, follow this procedure:

1. Open a terminal window and go to the case directory.

2. Decompose the case using: `decomposePar`.

3. Run (here N is the number of cores as it is defined in file "decomposeParDict"): `mpirun -np N passiveScalarLESBuoyantBoussinesqPimpleFoam -parallel`.

4. After the run has been completed, run this command for reconstructing the data from different processors: `reconstructPar`.

5. For calculating the vorticity and having it for post-processing part, run: `postprocess -func vorticity`.

The main software for post-processing of the OpenFOAM cases is called ParaFOAM. For activating that, use the following commands:

1. Run: `paraFoam`.

2. For post-processing the solution of processor N, run: `paraFoam -case processor`.

For monitoring the residuals during the runtime or at the end, use the following steps:

1. Open a terminal window and go to the case directory.

2. For generating a log file in the case folder, run: `passiveScalarLESBuoyantBoussinesqPimpleFoam > log &`

# References


[Ahmadi-Baloutaki and Aliabadi, 2021] Ahmadi-Baloutaki, M. and Aliabadi, A. A. (2021). A very large-eddy simulation model using a reductionist inlet turbulence generator and wall modeling for stable atmospheric boundary layers. *Fluid Dynamics*, 56(3):413–432.

[Aliabadi et al., 2017] Aliabadi, A. A., Krayenhoff, E. S., Nazarian, N., Chew, L. W., Armstrong, P. R., Afshari, A., and Norford, L. K. (2017). Effects of roof-edge roughness on air temperature and pollutant concentration in urban canyons. *Bound.-Lay. Meteorol.*, 164(2):249–279.

[Aliabadi et al., 2018] Aliabadi, A. A., Veriotes, N., and Pedro, G. (2018). A Very Large-Eddy Simulation (VLES) model for the investigation of the neutral atmospheric boundary layer. *J. Wind Eng. Ind. Aerodyn.*, 183:152–171.

[Kia et al., 2021] Kia, S., Flesch, T. K., Freeman, B. S., and Aliabadi, A. A. (2021). Atmospheric transport over open-pit mines: The effects of thermal stability and mine depth. *Journal of Wind Engineering and Industrial Aerodynamics*, 214:104677.