EPJ Web of Conferences **321**, 01008 (2025) https://doi.org/10.1051/epjconf/202532101008
APITECH-VII-2025

On developing a module for the stressFOAM solver of the OpenFOAM environment

Dmitry Chitalov*

South Urals Federal Research Centre of Mineralogy and Geoecology of the UB RAS, Russia, Chelyabinsk region, Miass, 456317, Ilmen reserve

Abstract. The present paper summarizes the study results on expanding capabilities of a graphical environment for the OpenFOAM program complex in operation with the stressFOAM solver. The OpenFOAM key disadvantage requiring further study is determined. The author defines urgency of the problem under study, analyzes the available solutions and their weak points. The study purposes and steps for their achievement are stated. The purpose of the stressFOAM solver and its basic application field are described. A stack of technologies to implement a graphical, software component of the application, data storage system is defined. Each selected technology of the stack is justified, the architecture of the software product under development is determined. The application development environment selected is presented. The structure and logic of the program module operation are shown in the corresponding diagrams with main program module components and an algorithm of its application for continuum mechanics problems. The main window of the module in the final phase of numerical simulation of a basic OpenFOAM problem related to the stressFOAM solver is viewed. Elements of the scientific study novelty are stated, expected practical value of the performed study for the end user is defined.

1 Introduction

The present work proceeds a cycle of the studies [1] developing a graphical environment for the OpenFOAM software environment [2]. This application permits engineers and researchers to perform numerical simulation of a wide range of continuum mechanics (CM) problems [3-7], including those with deformable solid bodies. The experimental results can be used in design departments of machine-building enterprises when designing products. Behavior patterns of real objects under environmental effects and processes are determined from final numerical models.

The key OpenFOAM disadvantage is the absence of a graphical user interface, i. e. screen forms used by the specialist to monitor the numerical experiment. It includes the phases of preprocessing (specify parameters corresponding to initial CM problem data), solving (to run numerical simulation), postprocessing (visualize the results and define a degree of their conformity to a real object or process). In each phase the specialist manually generates a case

* Corresponding author: cdi9@yandex.ru

structure, creates and fills internal files in with design parameters, starts the required OpenFOAM utilities and solvers. This approach presents a probability of errors and is rather labor-intensive.

International groups of researches are familiar with this problem. Software companies and independent developers tried to solve it. Their works resulted in some applications with a graphical user interface. Particularly, Visual-CFD, Salome, Helyx-OS software products became widespread. But the problem is still urgent due to fee-based technical support of the specified software products, necessity of purchasing a license, absence of detailed user documentation.

The presented study is focused on development and integration into the basic version of a native graphical interface [1] of a new program module that expands capabilities of the specialists in numerical simulations with OpenFOAM. The developed program module allows of reducing time spent by the specialists and minimize a probability of preparing a case with invalid input parameters. This program module also extends the OpenFOAM source code.

The application proposed by the author replaces a traditional method of the user operation with the stressFOAM solver. This is the OpenFOAM-integrated program permitting to perform a linear analysis of stresses for elastic solids and applying for stationary and nonstationary CM problems. The solver is a distributive component responsible for the process of numerical simulation of a definite CM problem.

2 Tools and methods

The present study is directed to replace a traditional approach when preparing a case for operation with the stressFOAM solver with the approach of applying graphical control tools. These tools include screen forms, menu bars and toolbars. The proposed module should provide a possibility of not only defining initial parameters of the CM problem via graphical tools, but adding automatically internal files with screen-defined parameters to the case.

The purpose achievement assumes performing the following set of tasks:

1) develop screen forms for internal files required in operation with the stressFOAM solver;
2) develop and code algorithms validating design parameters specified with screen form controls;
3) develop and code algorithms responsible for generation in the case directory of internal files with CM problem parameters;
4) develop and implement a mechanism of saving and recovering initial design parameters to enable their further editing;
5) create and integrate into the basic version a graphical environment of controls ensuring operation with the developed module.

2.1 StressFOAM solver purpose

As mentioned above, this is a standard program of the OpenFOAM distributive to simulate CM problems. The stressFOAM solver is designed for problems studying stresses to solid bodies. When preparing a case the program user has a possibilty of enabling or disabling the temperature influence option. If we talk about real numerical experiments, more attention is paid to isothermal mechanical stress, and temperature effects are not taken into account.

Before using the solver, the user builds a mesh model. This is a mandatory phase of the experiment related to the solution preprocessing. When building meshes, geometric models of bodies are determined, in this case we are talking about solids. Various types of meshes are used for CM problems, but more often block ones. In OpenFOAM such meshes are built using the standard blockMesh utility.

The traditional field of application of the stressFOAM solver is hydrodynamic problems, for example, the study of the interaction of sea bottom, water flows and metal structures. These problems are investigated in marine engineering, as well as in the design of underwater vehicles. Specialists identify possible variations in the characteristics of bodies, including hardness and strength. The obtained results are applied when designing products to ensure the required quality characteristics.

2.2 Development technologies

Each program product is implemented as a set of scripts. They describe the developed algorithms using expressions of a specific program language defined before creating a program product. In this phase, if necessary, the technology for implementing the graphical interface, data storage method, application development environment and other technologies are determined. The architecture of the future product is also determined in the development phase. In this case the program module will have a multi-layer architecture and function as a desktop software tool.

- Logic program language. Since the proposed module extends the capabilities of the basic graphical environment, the language used should be flexible and scalable, i.e., if necessary, allow of extending the application. The rate of language learning, ease of its syntax, and large community of specialists will also be important criteria for the developer. Based on the above requirements, it was decided to implement the logic of the program module using the high-level Python language.
- Graphical interface programming technology. It is important to use a library that supports syntax of the selected program language. At the same time, it should allow of creating familiar window interfaces and be accompanied by detailed documentation. One such library for Python is PyQt.
- Technology for implementing the data storage subsystem. All user-defined design parameters should be saved in a certain way and subsequently recovered. Relational databases will be optimal for data storage. SQLite databases are used for desktop program products. Python uses the standard sqlite3 module to operate with them.

Development environment. This is a window-based programming tool that provides not only coding, but also its debugging, testing, as well as running the entire project to assess its operability. For Python applications a popular development environment is PyCharm.

3 Results

3.1 Structure

The proposed program module includes a number of files containing a program code of scripts (Figure 1). This is not the final set of files, since the module assumes the expansion of functionality and connection of new blocks. The files are combined into internal directories based on their purposes. The root directory and primary executable file are created for the module.

The root directory in addition to the file being run (run.py) contains several nested directories: windows, forms, threads, functions. The first one contains window layouts for screen forms. In fact, we create environment for each form. In PyQt language it can be a widget window, dialog box, etc. In the second directory (forms) we already define the structure of controls for each form. These controls are fixed to the window, and we get a ready-made screen form. The third directory (threads) contains files with scripts for starting generation of computational meshes, running solvers that are directly responsible for numerical simulation, and starting utilities for visualizing the results. The fourth directory (functions) contains files with the code of service functions responsible, for example, for switching windows of the graphical environment, displaying service messages, etc.

3.2 Module logic

Operation with the presented module includes preprocessing, solving and postprocessing (Figure 2).

After running the module, the specialist specifies parameters of the computational mesh using the screen form and starts up one of the utilities for generating a mesh model, for example, blockMesh. Then the degree of compliance of the computational mesh with the stated requirements is determined by means of the ParaView visualization toolkit [8]. In case of a negative result the specialist can again proceed to edit the mesh parameters and perform regeneration. If the result is positive, he proceeds to determine the parameters of the CM problem and run the stressFOAM solver. In so doing, the case structure with the required internal files is formed. The results of the scientific experiment are also visualized using the ParaView platform and, if necessary, corrected (postprocessing).

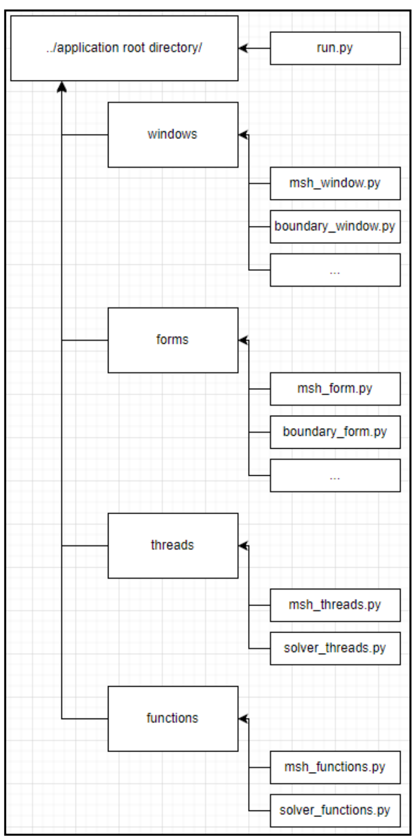

Fig. 1. Diagram of the module structure.

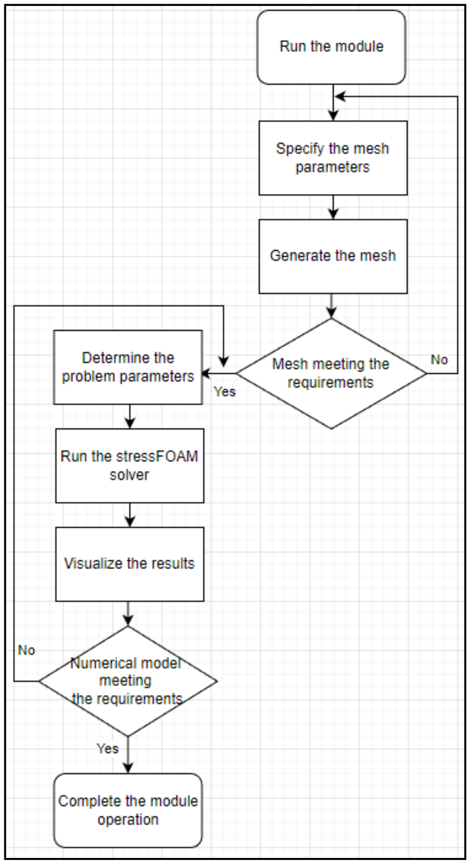

Fig. 2. Diagram of the module logic.

3.3 Checking the module operation

The proposed module is developed using free technologies and does not require purchasing a license. It extends the OpenFOAM source code and can be supplemented with new components. The current module version is available in the GitHub repository [9]. The application can be used by engineers of machine-building enterprises using OpenFOAM when designing products.

The module is applied for problems studying stresses to solid bodies. The module operation is tested on a classic CM problem included in the basic version of OpenFOAM. This is a plateHole problem where the solid is a square plate with a round hole in the center. The plate is loaded with a horizontal tension force on the right side. The view of the main application window after the experiment is completed is shown in Figure 3.

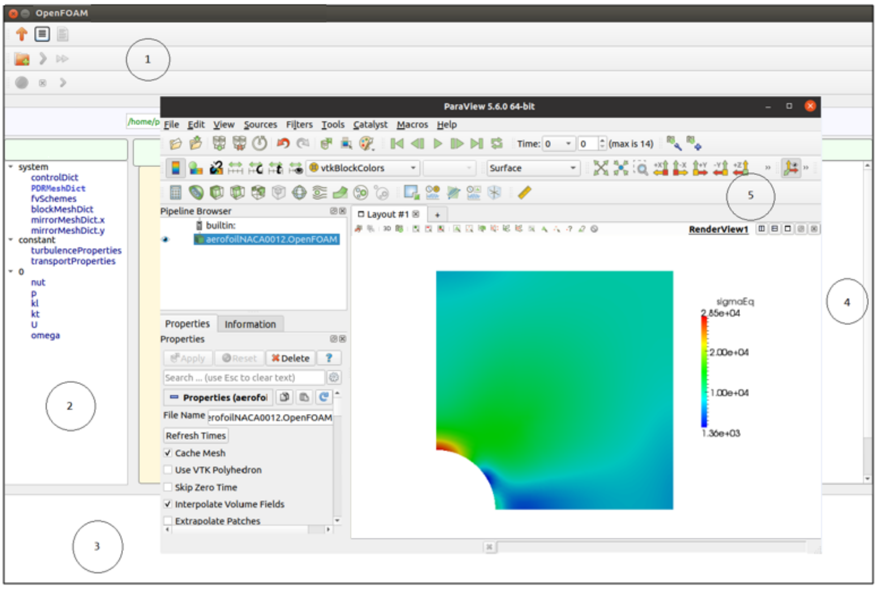

Fig. 3. Main window of the module after the experiment is completed, where 1 is a simulation control panel, 2 is case components, 3 is a status bar, 4 is a parameter input panel, 5 is the ParaView window.

4 Discussion

Existing analog applications have a rather redundant interface. Their learning requires considerable time, detailed documentation, and often the help of specialists. The graphical environment proposed by the author differs in that it is implemented according to the modular principle, i.e. additional modules are connected to expand the capabilities. In this regard, the interface of the presented product can be described as intuitive, requiring no prior preparation.

In addition, due to the modular principle we can talk about a great potential for further researches in this area. It is planned to develop and integrate new modules into the basic graphical environment, which will increase the capabilities of specialists in preprocessing (expanding a list of available utilities for generating computational meshes), solving (access to more solvers for simulating new CM problems), prostprocessing (expanding a list of utilities for studying the results).

5 Conclusion

The result of the study carried out by the author is a program module that expands the capabilities of specialists in numerical simulation of CM problems through the OpenFOAM complex. The module provides the user with graphical (interface) and software (scripts) tools ensuring the creation of a computational case for experiments in the field of solid stress analysis. In OpenFOAM similar CM problems are simulated with the stressFOAM solver.

The scientific novelty of the study is further formulated as a list of approaches that have been proposed and implemented in the graphical and software component of the module:

1. separating the source code of screen form windows from the code of the form controls. This allows you to place form components in other types of windows, if

necessary. The files with the code of screen form elements also contain scripts ensuring creation of internal files of the case with the user-entered parameters;

2. using bash scripts to run OpenFOAM utilities and solvers. Instead of the traditional manual running via the command line the specialist uses graphical running (interface). At the same time the utilities and solvers run due to bash scripts. Thus, the user does not need to know complex OpenFOAM commands. All the work is carried out through the module tools;
3. mechanism for checking the types of data entered through screen forms. If an incorrect value is entered, the numerical experiment may end with an error, therefore, a mechanism is proposed to verify these values before generating the computational mesh and running numerical simulation;
4. mechanism for checking the case complete set. With the traditional approach to numerical simulation based on OpenFOAM, a specialist needs to have sufficient experience and knowledge necessary to work with this program complex. There is a possibility of creating an incorrect case structure, when not all internal files with design parameters will be added. The proposed mechanism monitors the case structure, so each phase of the numerical experiment can be carried out only if all necessary files are available.

The presented module replaces the traditional, time-consuming and error-prone manual approach of creating a computational case and using the command line with a more convenient and efficient one when all the steps of a numerical experiment are carried out using usual screen forms. In addition to improving user convenience, the expected practical value is to reduce time spent by a specialist.

References

1. D.I. Chitalov, S.T. Kalashnikov, *Development of User Interface for OpenFOAM Software Environment Used in Design and Technological Subdivisions of Machine-Building Enterprises* in: Global Smart Industry Conference (GloSIC) (Chelyabinsk*),* pp. 1-7 (2018).
2. OpenFOAM. The open source CFD toolbox (2024). https://www.openfoam.com/
3. A. Etrati, I.A. Frigaard, Physics of Fluids **30(2)**, 022107 (2018).
4. S. Galindo-Lopez, F. Salehi, M.J. Cleary, A.R. Masri, G. Neuber, O.T. Stein, Computers and Fluids **172**, 410-425 (2018).
5. H. Jelawy, S. Kaczmarczyk, D. AlKhafaji, S. Mirhadizadeh, R. Lewis, M. Cross, *A Computational Investigation of a Turbulent Flow over a Backward Facing Step with OpenFOAM* 9th International Conference on Developments in eSystems Engineering (DeSE) (Liverpool), pp. 301-307 (2016).
6. N. Kyriazis, P. Koukouvinis, M. Gavaises, Advances in Colloid and Interface Science **260**, 46-64 (2018).
7. M. Mohseni, P.T. Esperanca, S.H. Sphaier, Applied Ocean Research **79**, 228-252 (2018).
8. ParaView (2024). https://www.paraview.org/
9. *OpenFOAM_GUI* (2007). https://github.com/DmitryChitalov/OpenFOAM_GUI