Development of an application with a graphical user interface (GUI) to compute in parallel in the OpenFOAM environment

Dmitri I Chitalov
South Urals Federal Research Centre of Mineralogy and Geoecology of the UB RAS, Miass, Ilmen reserve, 456317, Chelyabinsk region, Russia
E-mail: cdi9@yandex.ru

Abstract. Means of man-computer interaction have passed a complicated path of development. Nowadays they look like visually easily understandable interfaces with graphical elements on a monitor screen. Interfaces allow forming initial data (ID) for a program and visualizing results. A major part of software has in-built graphical environments to interact with a user. Some open source environments, e.g., the OpenFOAM used for numerical simulation of continuum mechanics (CM) problems, have no in-built graphical means and are designed to operate with a command line. Even though some developers of software environment have provided their decisions of the problem, the problem remains acute as the available decisions are not lacking in drawbacks. The paper focuses on the description of features of the creation of an original graphical environment to operate with OpenFOAM for CM problems computing in parallel. With this aim in mind a list of required tasks is made out and the needed tools are designated: programming language Python 3.5, framework of graphical elements PyQt5 and development environment PyCharm. Diagrams describing structure and behavior of the developed application are worked out. The results of development and testing of the application are shown through the example of one of standard training tasks of the OpenFOAM distributive. Statements that have scientific novelty of the development are formulated, the practical use of the development is defined. A balance of work is stricken and prospects of further study in the direction are identified.

1. Introduction
A personal computer is an inalienable element in man's everyday life. Modern computing devices have the following interaction classes: personal software (applications) and personal platforms (OS, programming languages, hardware).

The use of the first computing devices was full of complications due to lack of suitable and clear means for user/computer interaction. The first GUI was developed in 1973 in the framework of the Xerox Alto project – an ancestor of all personal computers. The proposed graphical environment was notable for easy operation and in that time had a menu, buttons, primitive windows and even a mouse cursor capable to highlight objects. With the appearance of Xerox Star OS in 1981 the existing GUI was upgraded to a version in many aspects identical to the available now.

The paper is devoted to features of the development of an application used to realize man/computer interaction: GUI development and description of its behavior. GUI is a kit of objects shown on a display with which a user enters ID and gets the results of their processing. The paper continues the research described in the papers [1-2].
Each year a lot of new applications have been developed to solve different problems, including those for computation, but not all of them have a convenient and clear user interface. OpenFOAM [3] environment is one of such applications not stipulating for any GUI. OpenFOAM is an open source tool to make calculations in the CM field on the basis of numerical methods. It also assists in creating digital models that simulate processes (problems) in such CM subdisciplines as gas and hydrodynamics, mechanics of a deformable solid body, etc. An example of simulation of a CM problem is described in [4].

OpenFOAM is well-reputed among design and technological subdivisions of enterprises of various branches of machine-building: automobile, engine, rocket building; space industry. Among the OpenFOAM users are the following machine-building giants: Audi, Volvo, Toyota, Volkswagen, Shell Oil.

The first OpenFOAM version was issued in 2004 and at the beginning users used only a traditional methodology to interact with the environment. Later on a series of foreign companies started projects on the development of fully-featured graphical interfaces for OpenFOAM. The most popular among them are Salome [5], Visual-CFD [6], Helyx-OS [7] that permit going through all phases of creating digital models of CM problems on the basis of OpenFOAM: pre-processing, running, post-processing.

The created GUIs to interact with OpenFOAM caused a mixed reaction of engineers and researchers. On one hand, users got the possibility not to utilize a time-consuming traditional methodology for numerical simulation, on the other hand, drawbacks were revealed in each of the proposed graphical environments due to which users were not ready to shift from the traditional interface to the window one. The following drawbacks were noted: obligatory payment for a license and consulting services, absence of detailed documentation on products. Thus, the international community of researchers is still interested in the creation of a graphical interface for OpenFOAM.

2. Urgency of an Issue
Before starting the development of own graphical environment for OpenFOAM the authors have analyzed functionalities and features in utilization of some well-known analogs. The comparison was made in several parameters: obligatory payment for a license on a software product, technical support and consulting services, the presence of training documentation. The measurement results are given in table 1.

Table 1. GUI for numerical simulation on the basis of OpenFOAM.

| Name         | License                  | Technical support         | Training documentation                  |
|--------------|--------------------------|---------------------------|-----------------------------------------|
| Salome       | Open source software     | Provided, to be paid       | Provided                                |
| Visual-CFD   | Demo version for 30 days  | Provided, to be paid       | Only basic description of capabilities  |
| Helyx-OS     | Open source software     | Provided, to be paid       | Only basic instruction for users         |

3. Setting goals and tasks
The study purpose is to develop graphical environments for OpenFOAM for computing in parallel (OpenFOAM_decompose_GUI). To answer the purpose, the authors have studied official documentation on OpenFOAM 6.0 and the description of decisions of training problems [8, 9]. A described graphical environment should be a separate application, each phase of numerical simulation should be realized through OpenFOAM: pre-processing (definition and inputting parameters of a CM problem), running (numerical simulation of the CM problem), post-processing (visualization of results of numerical simulation). Some features and examples of computing CM problems on the basis of
OpenFOAM are provided in [10-13]. To perform each phase of the numerical simulation one should develop the GUI structure and define its behavior.

GUI structure:

- A primary window of the application.
- A panel to control basic setting up of numerical simulation.
- The panel to control the preparation of meshes should have a button to open a window for specifying options of operations with meshes, a window to start generation of meshes, a button to start visualization of results.
- The panel to control basic setting up of numerical simulation should have buttons to start numerical simulation of a CM problem, to stop the process and visualize the results.
- The block of the case tree should realize output in a tree from in a separate window.
- The block for editing parameters of the case utilities should be a window into which an electronic screen corresponding to a utility should be loaded.
- The block to visualize the content of utilities should be a window in which the content of each utility of the case should be displayed.
- The panel for signal messages should be a special window to inform a user about the status and results of actions executed by him during the numerical simulation.
- A block to display paths to the case directory and the case mesh file (or directory with files of the mesh parameters).

GUI behavior is featured by possibilities to:

- Create a new case and change parameters of the existing case;
- Generate a mesh on the basis of msh-files created with the third-party software and generate meshes with the OpenFOAM means (with built-in utilities blockMesh and snappyHexMesh);
- Start and stop numerical simulation of CM tasks, visualize the simulation results;
- Warn a user in the course of numerical simulation during pre-processing, running a CM problem, visualization of results;
- Validate data input through electronic screens and check the presence of utilities before the numerical simulation is started and the results are displayed.

4. Theoretical part

4.1. Selecting tools for development

The application behavior is decided to be realized with the Python language which has reinforced its positions as of 2018 and is reported to be the most popular tool to describe the behavior of desktop and web-applications [14]. An important advantage of the Python language is a minimum threshold for logging-in, i.e., minimum programming knowledge is enough for getting acquainted with it. Python features easy, user-friendly syntax [15]. Specialized frameworks, e.g., PyQt [16], are used to describe GUI elements. A full list of tools is summarized in table 2.

Table 2. A list of tools for developing the application.

| Programming language | Framework for graphical elements | Environment for development | Programming language |
|-----------------------|----------------------------------|-----------------------------|----------------------|
| Python 3.5            | PyQt5                            | PyCharm 2.4                 | Python 3.5           |

5. Practical part
5.1. Results of study
Figure 1-4 presents images of the application main window (GUI) and separate interface blocks. The application is tested based on training task depthCharge3D that is included in a set of training examples of the OpenFOAM standard distributive.

**Figure 1.** A main window, where: 1-application main window, 2- panel to control basic setting up of numerical simulation, 3-panel to control the preparation of meshes, 4-panel to control the progress of numerical simulation, 5-window for preparation of a case in OpenFOAM.

**Figure 2.** A main window, where: 6-block of the case tree, 7-panel for signal messages, 8-window for specifying options of operations with meshes.

**Figure 3.** A main window in the completion of creation of blockMeshDict file with the mesh parameters, where: 9-block to display paths to the case directory and the case mesh file, 10-block for editing parameters of the case utilities, 11-block to visualize the content of utilities.
5.2. Scientific novelty of the study

The authors have analyzed potential difficulties of the structural, logical and algorithmical types. To overcome difficulties, they proposed approaches with features of scientific novelty:

- A methodology for the creation of a set of forms for editing parameters of files corresponding to mesh models, i.e., files blockMeshDict and snappyHexMeshDict.
- A mechanism of serialization of parameters of meshes [18]. The authors propose the mechanism for reservation of parameters of mesh models as a kit of utilities, where each utility contains parameters of a mesh model stored in a certain block of the main file of the mesh model.
- A method used to create tables of database for files of the case parameters. The authors proposed a mechanism to save the CM problem parameters defined in the framework of a modeled case.
- A method to generate bash-scripts and start them up as child processes with the Python programming language [19] permits escaping traditional manual start-up of console commands for preparation of meshes and executing numerical modeling on the basis of OpenFOAM.
- A method to install validators to validate the type of inputting data permits not to input incorrect data, as a result, to reveal errors in formulation of numerical simulation.
- A method to check the completeness of the case utilities. E.g., a user can generate a mesh before preparation of a file corresponding to the mesh type, i.e., file blockMeshDict or snappyHexMeshDict.

5.3. Practical relevance of the study

The created by the authors OpenFOAM_decompose_GUI application permits to replace a complicated time-consuming method by an alternative one and go through all phases of numerical simulation centrally via a common window GUI. This gives several advantages:

- Saving time. The application creates a case with enclosed catalogs and utilities with parameters of a CM problem; it also starts up utilities of pre-processing, running and post-processing.
- Minimum mistakes. The application checks the correctness of inputting data and the completeness of files of directories of a case with validators before running and visualization of results.
6. Conclusions
The paper presents an original graphical environment for numerical simulation on the basis of OpenFOAM realized with programming language Python 3.5 and framework of graphical elements PyQt5. The application permits performing numerical simulation with running in parallel, thus, saving computer resources and time for numerical simulation. The application is free, has user-friendly interface, no payment for technical support. The application is available at the GitHub service and can be used by a team of researchers and machine-building enterprises that apply OpenFOAM in their activities.

References
[1] Chitalov D I and Kalashnikov S T 2018 Development of User Interface for OpenFOAM Software Environment Used in Design and Technological Subdivisions of Machine-Building Enterprises Global Smart Industry Conference (GloSIC) (Chelyabinsk) pp. 1–7
[2] Chitalov D I and Kalashnikov S T 2019 Development of a computational mesh generator for the OpenFOAM software environment used at industrial enterprises IOP Conf. Ser.: Mater. Sci. Eng. 537 032048
[3] OpenFOAM. The open source CFD toolbox. [Online]. Available: https://www.openfoam.com/
[4] Etrati A and Frigaard I A 2018 Viscosity effects in density-stable miscible displacement flows: Experiments and simulations Physics of Fluids, 30(12)
[5] Salome. The Open Source Integration Platform for Numerical Simulation. [Online]. Available: http://www.salome-platform.org
[6] Visual-CFD for OpenFOAM. [Online]. Available: https://www.esi-group.com/software-solutions/virtual-environment/cfd-multiphysics/visual-cfd-openfoam
[7] HELYX-OS. The market leading open-source GUI for OpenFOAM. [Online]. Available: http://engys.com/products/helyx-os
[8] OpenFOAM. User Guide. [Online]. Available: http://foam.sourceforge.net/docs/Guides-a4/OpenFOAMUserGuide-A4.pdf
[9] OpenFOAM. Tutorial Guide. [Online]. Available: https://www.openfoam.com/documentation/tutorial-guide/index.php
[10] Galindo-Lopez S, Salehi F, Cleary M J, Masri A R, Neuber G and Stein O T 2018 A stochastic multiple mapping conditioning computational model in OpenFOAM for turbulent combustion Computers and Fluids 172 410-25
[11] Kyriazis N, Koukouvinis P, and Gavaïses M 2018 Modelling cavitation during drop impact on solid surfaces Advances in Colloid and Interface Science 260 46-64
[12] Al-Jelawy H, Kaczmarczyk S, AlKhañaji D, Mirhadjadeh S, Lewis R, and Cross M 2016 A Computational Investigation of a Turbulent Flow over a Backward Facing Step with OpenFOAM 2016 9th International Conference on Developments in eSystems Engineering (DeSE), Liverpool pp 301-7
[13] Mohseni M, Esperanca P T, and Sphaier S H 2018 Numerical study of wave run-up on a fixed and vertical surface-piercing cylinder subjected to regular, non-breaking waves using OpenFOAM Applied Ocean Research 79 228-52
[14] TIOBE index for January 2019. [Online]. Available: https://www.tiobe.com/tiobe-index/
[15] Python 3.5 documentation. [Online]. Available: https://docs.python.org/3.5/
[16] PyQt5 Reference Guide. [Online]. Available: http://pyqt.sourceforge.net/Docs/PyQt5/
[17] OpenFOAM_decompose_GUI. [Online]. Available: https://github.com/DmitryChitalov/OpenFOAM_decompose_GUI
[18] Advanced bash-scripting guide. [Online]. Available: https://www.tldp.org/LDP/abs/html/