

УДК 004.512
DOI: 10.15827/0236-235X.127.433-443

Дата подачи статьи: 27.03.19
2019. Т. 32. № 3. С. 433–443

Разработка графической оболочки для параллельных расчетов на базе платформы OpenFOAM

Д.И. Читалов¹, младший научный сотрудник, cdi9@yandex.ru

¹ Южно-Уральский федеральный научный центр минералогии и геоэкологии УрО РАН, г. Миасс, Ильменский заповедник, 456317, Россия

Графические интерфейсы являются важнейшим элементом человеко-компьютерного взаимодействия. Они обеспечивают формирование входных данных для программ и визуализацию результатов. Коммерческое ПО преимущественно поставляется со встроенными графическими средствами взаимодействия. При этом некоторые открытые программные решения, в частности, платформа OpenFOAM, лишены встроенных средств взаимодействия. Данная проблема сохраняет свою актуальность, поскольку существующие графические интерфейсы имеют недостатки.

В настоящей работе представлен поэтапный процесс разработки графической оболочки для реализации взаимодействия пользователя с платформой OpenFOAM – постановки численных экспериментов применительно к задачам механики сплошных сред в режиме параллельного выполнения. Для достижения поставленной цели определен список задач и необходимых инструментов: язык программирования Python 3.5, фреймворк описания элементов интерфейса PyQt5, интегрированная среда разработки PyCharm.

Приведены диаграммы, демонстрирующие взаимосвязь модулей программы, и механизм работы программы. Сформулированы результаты разработки и тестирования приложения на примере проекта одной из стандартных задач механики сплошных сред, входящих в дистрибутив платформы OpenFOAM. Сформулирована научная новизна исследования, в частности, сериализация параметров численного эксперимента с помощью модуля Pickle языка Python и таблиц БД в формате SQLite, возможность создания различных версий файлов с параметрами задач механики сплошных сред, возможность запуска консольных команд с помощью bash-скриптов. Определены практическая ценность работы и дальнейшие перспективы.

Ключевые слова: численное моделирование, механика сплошных сред, параллельные вычисления, OpenFOAM, открытое ПО, графический интерфейс пользователя, язык программирования Python, библиотека PyQt, СУБД SQLite, bash-скриптинг.

Персональный компьютер является неотъемлемым элементом жизни человека. Использование первых вычислительных устройств было сопряжено с большими сложностями из-за отсутствия удобных и понятных средств взаимодействия между пользователем и самим устройством. Первый графический пользовательский интерфейс (GUI) был разработан в 1973 году в рамках проекта Xerox Alto – родоначальника всех персональных компьютеров. Предложенная графическая оболочка отличалась простотой, но уже предусматривала наличие меню, кнопок, примитивных окон и даже курсора мыши с возможностью выделения объектов. С появлением в 1981 году ОС Xerox Star существующий GUI был усовершенствован до версии, которая во многом идентична существующим [1].

Настоящая статья посвящена особенностям проектирования программного средства для реализации взаимодействия человека и компь-

ютера: проектирования GUI и описания логики его работы. GUI представляет собой набор объектов, отображаемых на дисплее, через которые пользователь вводит данные и получает результаты их обработки.

Ежегодно создается множество новых программных средств для решения различных задач, в том числе проведения вычислений. Однако не все они обладают удобным и понятным интерфейсом пользователя, поэтому требуются специальные навыки и значительные ресурсы рабочего времени на освоение. Одним из таких программных приложений является платформа OpenFOAM [2], не предусматривающая наличие GUI. Платформа OpenFOAM является свободно распространяемым инструментарием для проведения расчетов в области механики сплошных сред (МСС) посредством численных методов. При этом создаются цифровые модели, имитирующие процессы (задачи), связанные с такими разделами МСС, как

газо- и гидродинамика, механика деформируемого твердого тела и др. Пример численного исследования для задачи МСС представлен в работе [3].

Платформа OpenFOAM успешно зарекомендовала себя в конструкторских и технологических подразделениях предприятий различных отраслей машиностроения: автомобилестроения, двигателестроения, ракетостроения, космического строения. Среди пользователей OpenFOAM представлены крупнейшие гиганты машиностроения: Audi, Volvo, Toyota, Volkswagen, Shell Oil.

Платформа OpenFOAM предусматривает использование только командной строки и требует от специалиста знания команд, предназначенных для запуска утилит-команд OpenFOAM, необходимых для управления численными исследованиями, а также структуры каждого расчетного случая (задачи МСС) и свойств задач МСС в служебных файлах. При существующем подходе специалисты вынуждены работать с расчетным случаем через командную строку и вручную редактировать служебные файлы. По их отзывам, указанный подход весьма неудобен и требует существенных временных затрат.

Альтернативой данному подходу является классический оконный графический интерфейс, благодаря которому численный эксперимент на основе платформы OpenFOAM будет осуществляться с применением окон и элементов управления: кнопок, выпадающих списков, текстовых полей, фреймов и т.д.

Первая версия OpenFOAM была выпущена в 2004 году, и первоначально специалисты для взаимодействия с данной платформой использовали только традиционный подход. Позднее некоторые иностранные IT-компании разработали собственные графические оболочки для OpenFOAM. Среди них наибольшую известность получили программные продукты Salome [4], Visual-CFD [5], Helyx-OS [6], которые позволяют на базе платформы OpenFOAM выполнять все этапы построения цифровых моделей задач МСС – препроцессинг, решение, постпроцессинг.

Создание перечисленных GUI для взаимодействия с OpenFOAM вызвало неоднозначную реакцию специалистов. С одной стороны, пользователи получили возможность отказаться от трудоемкого традиционного проведения численного моделирования, а с другой – в каждой из предложенных графических оболочек были обнаружены недостатки, из-за кото-

рых не все пользователи оказались готовыми к переходу от традиционного подхода к оконному интерфейсу. Среди недостатков были отмечены необходимость оплаты лицензии и консультационных услуг, а также отсутствие полноценной документации по продуктам. Таким образом, у исследователей сохраняется интерес к созданию графической оболочки для платформы OpenFOAM.

Актуальность проблемы

Перед созданием оригинального программного продукта были проанализированы функциональные возможности и особенности нескольких наиболее известных аналогов. Результаты сравнения приведены в таблице.

GUI для постановки численных экспериментов на базе платформы OpenFOAM GUI for numerical experiments based on the OpenFOAM platform

Название	Лицензия	Техническая поддержка	Обучающая документация
Salome	Свободно распространяемое ПО	Существует, платная	Предусмотрена
Visual-CFD	Бесплатная версия на 30 дней	Существует, платная	Только базовое описание возможностей
Helyx-OS	Свободно распространяемое ПО	Существует, платная	Только базовая инструкция для пользователей

На основе информации, представленной в таблице, можно сделать вывод, что использование приведенных графических оболочек на предприятии требует выделения финансовых средств на оплату консультационных услуг по внедрению и освоению продукта. Обучающая документация содержит лишь базовую информацию о продукте и его основных возможностях, так как разработчики представленных программных решений заинтересованы в том, чтобы потенциальные заказчики оплачивали техническую поддержку.

Не каждое предприятие располагает необходимыми финансовыми ресурсами на приобретение коммерческого ПО и оплату технической поддержки. Такая возможность существует прежде всего у крупных промышленных

компаний. Небольшие предприятия вынуждены переходить на использование открытых программных средств, имеющих необходимую функциональность, понятных в применении и не требующих оплаты технической поддержки. В случае с платформой OpenFOAM такое программное средство отсутствует, поэтому обозначенная проблема сохраняет свою актуальность.

В существующих коммерческих программных продуктах предусмотрена возможность параллельного решения задач МСС при одновременной загрузке до тысячи вычислительных ядер. Однако высокая стоимость лицензий на использование вычислительных ядер вынуждает исследователей переходить на свободно распространяемое ПО. При этом открывается доступ к исходному коду без каких-либо ограничений по использованию программных решений, в том числе и при параллельных расчетах. Одним из таких решений, предусматривающих проведение параллельных расчетов через декомпозицию расчетной области, является платформа OpenFOAM.

Данный механизм уже встроен в платформу OpenFOAM и не требует от пользователя доработки программного кода. Параллельные расчеты осуществляются в несколько этапов, начиная с указания количества подобластей, на которое необходимо разделить расчетную область. Также требуется указание способа разделения (ручного или автоматического) для наиболее эффективного распределения нагрузки между процессорами.

Далее происходит распределение исходных данных между процессорами. После завершения решателем своей работы требуются сборка итогового решения, а затем обработка и отображение результатов. В итоге с увеличением количества ядер снижается расчетное время. При этом благодаря открытости платформы OpenFOAM не требуется лицензия на использование в расчетах дополнительных ядер. Это доказывает эффективность OpenFOAM в рамках проведения параллельных расчетов без необходимости дополнительных операций. Важнейший элемент – запуск команды `decomposePar` для декомпозиции расчетной области, которая составляет несколько секунд.

Постановка целей и задач

Первая версия графической оболочки для платформы OpenFOAM была разработана в 2016 году [7]. Она имела ограниченный функ-

ционал – возможность использования только применительно к решателю `rhoCentralFoam` (невязкий сжимаемый поток) платформы OpenFOAM. Кроме того, первая версия не предусматривала выполнение параллельных вычислений.

Цель настоящей работы – создание графической оболочки (GUI) для платформы OpenFOAM с возможностью параллельных расчетов (OpenFOAM decompose GUI) и применения для широкого круга решателей OpenFOAM: `icoFoam` (несжимаемый ламинарный поток), `simpleFoam` (стационарный несжимаемый турбулентный поток), `pisoFoam` (нестационарный несжимаемый турбулентный поток), `sonicFoam` (нестационарный сжимаемый турбулентный поток).

Для реализации поставленной цели были изучены официальная документация по OpenFOAM 6.0 и решения учебных задач [8, 9]. Описываемая графическая оболочка должна быть реализована в виде отдельного программного приложения и предусматривать выполнение каждого из этапов численного эксперимента с помощью платформы OpenFOAM: препроцессинг (определение и установка параметров задачи МСС), решение (численное моделирование задачи МСС), постпроцессинг (визуализация результатов численного эксперимента). Особенности и примеры решения задач МСС на базе платформы OpenFOAM приведены в работах [10–13]. Для выполнения каждого из этапов численного эксперимента необходимо спроектировать структуру GUI и определить логику его работы.

Представим структуру GUI.

1. Главное окно программы. Должно включать семь интерфейсных блоков: панель управления базовыми настройками численного эксперимента, панель управления подготовкой расчетных сеток, панель управления ходом численного эксперимента, блок дерева расчетного случая, блок редактирования параметров служебных файлов расчетного случая, блок отображения содержимого служебных файлов, панель отображения служебных сообщений.

2. Панель управления базовыми настройками численного эксперимента. Должна содержать кнопку открытия окна подготовки расчетного случая платформы OpenFOAM, кнопку открытия окна выбора языковой версии интерфейса и кнопку открытия универсального окна для создания файлов с начальными условиями численного эксперимента, то есть файлов для директории «0» расчетного случая. Окно под-

готовки расчетного случая должно предусматривать радиокнопки для выбора нужной опции: создание нового расчетного случая или выбор существующего. Также в окне необходимо предусмотреть поле ввода для указания названия расчетного случая и кнопку для открытия стандартного окна выбора директории расчетного случая. Окно выбора языковой версии интерфейса должно содержать радиокнопки для выбора языковой версии интерфейса.

3. Панель управления подготовкой моделей сеток. В этой панели необходимо реализовать кнопку открытия окна для указания опций работы с расчетными сетками, кнопку запуска формирования моделей сеток, кнопку запуска отображения результатов. Окно указания опций работы с расчетными сетками должно содержать элементы управления для выбора механизма генерации сетки – импорт внешнего файла, созданного посредством стороннего ПО, или создание сетки средствами OpenFOAM (с помощью утилит blockMesh и snappyHexMesh). Механизм построения моделей сеток на основе указанных утилит описан в работах [14, 15].

4. Панель управления ходом численного эксперимента. Должна содержать кнопки запуска численного эксперимента, его останова и визуализации итоговых результатов.

5. Блок дерева расчетного случая. Реализует вывод структуры проекта задачи МСС в древовидном формате в отдельном окне. По выделению нужной ветви дерева (нужного служебного файла расчетного случая) в блоке редактирования параметров служебных файлов расчетного случая должна загружаться соответствующая экранная форма.

6. Блок редактирования параметров служебных файлов расчетного случая. Реализуется в виде окна, в которое должна загружаться экранная форма, соответствующая служебному файлу. В каждой форме, в соответствии со структурой служебного файла, необходимо расположить соответствующие элементы управления.

7. Блок отображения содержимого служебных файлов. Реализуется в виде окна, в котором должно отображаться содержимое каждого служебного файла расчетного случая.

8. Панель отображения служебных сообщений. Специальное окно для информирования пользователя о ходе численного эксперимента.

9. Блок отображения путей до проекта расчетного случая и файла расчетной сетки (или

директории со служебными файлами параметров расчетной сетки).

Логика работы GUI:

- возможность подготовки нового расчетного случая и изменение параметров существующего;

- возможность генерации моделей сеток на основе msh-файлов, подготовленных посредством сторонних программных продуктов, и генерации моделей сеток средствами OpenFOAM (с помощью встроенных утилит blockMesh и snappyHexMesh);

- запуск и останов численного эксперимента, визуализация результатов моделирования;

- оповещение пользователя о ходе выполнения этапов численного эксперимента;

- валидация данных, вводимых через экранные формы, и проверка комплектности служебных файлов перед запуском численного эксперимента и визуализацией результатов;

- поддержка в интерфейсе программы различных языков.

В процессе разработки программы проанализированы возможные сложности структурного, логического и алгоритмического характера. Для их решения предложено использование следующих подходов, составляющих научную новизну работы.

- Создание набора форм для редактирования параметров файлов, соответствующих сеточным моделям, то есть файлов blockMeshDict и snappyHexMeshDict. В данных файлах параметры сеток разделены по блокам. Каждому блоку соответствует отдельная форма набора. Заполнение форм осуществляется в строгом порядке, причем структура каждой из форм зависит от содержимого предыдущей формы. Тем самым упрощается подготовка сеточной модели пользователем.

- Сериализация параметров расчетных сеток. Предложен механизм консервирования параметров сеточных моделей в виде набора служебных файлов, где каждый файл содержит параметры сеточной модели, хранящиеся в определенном блоке главного файла сеточной модели, например, файла blockMeshDict или snappyHexMeshDict. Благодаря такому подходу параметры сеточной модели консервируются и могут быть восстановлены для последующего редактирования.

- Создание таблиц БД для файлов параметров расчетного случая. Предложен механизм сохранения параметров задачи МСС, определяемых в рамках моделируемого расчет-

ного случая. Эти параметры записываются в соответствующие файлы расчетного случая, например, в `controlDict`, `thermophysicalProperties`, `decomposeParDict`. Согласно предложенному механизму, для каждого из расчетных случаев создается файл БД в формате SQLite, где каждая из таблиц соответствует определенному файлу, то есть содержит структуру полей и соответствующих им значений (параметров задачи МСС). Благодаря этому механизму параметры задачи МСС могут быть не только сохранены в базе, но и загружены в соответствующую экранную форму при необходимости их редактирования.

- Генерация `bash`-скриптов и запуск их в качестве дочерних процессов средствами языка программирования Python [16]. Использование данного подхода позволяет отказаться от привычного ручного запуска консольных команд по подготовке расчетных сеток и проведению численного моделирования на базе платформы OpenFOAM.

- Установка валидаторов для проверки типа вводимых данных. Позволяет избежать ввода некорректных данных, а значит, и ошибок в процессе постановки численного эксперимента.

- Проверка комплектности служебных файлов расчетного случая. Например, пользователь не может выполнить генерацию расчетной сетки до тех пор, пока не подготовит файл, соответствующий данному типу сеточной модели, то есть файл `blockMeshDict` или `snappyHexMeshDict`.

- Реализация возможности создания служебных файлов с различными начальными условиями численного эксперимента. Данный подход позволяет в рамках одного расчетного случая (одной задачи МСС) создать любое количество служебных файлов с начальными параметрами, то есть у пользователя есть возможность выполнить численное моделирование при различных начальных условиях. Речь идет о файлах директории «0» расчетного случая OpenFOAM: p (давление), T (температура) и U (скорость). Согласно предложенному подходу, пользователь может создать сразу несколько вариантов этих файлов с различными параметрами.

Выбор инструментария разработки

Разрабатываемая графическая оболочка является программным средством, то есть набором файлов с программным кодом. Для его

написания применяется специальный инструментарий, который включает язык программирования и среду разработки. Разрабатываемый интерфейс предполагается реализовать в виде настольного программного приложения. При этом применяются различные высокоуровневые языки программирования, например, C/C++, Java и Python. Они входят в список наиболее популярных языков программирования по рейтингу TIOBE [17].

Принято решение реализовать логику работы программы с помощью языка Python, который по итогам 2018 года укрепил свои позиции и считается одним из наиболее популярных средств описания логики работы настольных и веб-приложений. Это мощный язык программирования, предусматривающий использование стандартных типов данных, а также поддерживающий многочисленные библиотеки для решения широкого перечня задач программирования. Кроме того, важным преимуществом языка Python является минимальный порог входа, то есть достаточно минимальных знаний в программировании для его освоения. Python отличается простым, интуитивно понятным синтаксисом [18].

Для описания элементов графических интерфейсов применяются специализированные фреймворки, например PyQt [19]. Данный фреймворк оптимален для описания внешнего вида интерфейсов настольных программ. Он содержит все необходимые средства для реализации классических оконных интерфейсов с привлекательным дизайном.

Еще одним важным аспектом определения инструментария является выбор среды разработки для написания, отладки и запуска программного кода. В последние годы среди Python-разработчиков пользуется популярностью интегрированная среда разработки PyCharm. Она позволяет для каждого проекта определить нужную версию интерпретатора Python, а также в режиме реального времени установить необходимые пакеты и расширения, в том числе PyQt.

Полный перечень применяемого инструментария: язык программирования Python 3.5, фреймворк графических элементов PyQt5, среда разработки PyCharm 2.4 (версия Community Edition). Каждый из инструментов открытый, то есть не требует приобретения лицензии. Версия Community среды разработки PyCharm является специальной бесплатной версией с ограниченным, но достаточным для большинства проектов функционалом.

Для использования графической оболочки необходимо следующее: ОС – Linux, программная среда – OpenFOAM 6.0, среда визуализации результатов – ParaView.

Среда визуализации результатов ParaView – это специализированная открытая программа, которая предоставляет возможность исследователям выполнять визуализацию (отображение) результатов численного эксперимента [20].

Структура и логика работы приложения

Приложение OpenFOAM_decompose_GUI реализовано по модульному принципу, в соответствии с которым за реализацию определенной функциональности отвечает определенный блок кода, помещенный в файл с расширением .py (Python-модуль). Набор модулей формирует общую структуру программы. Структура представлена на рисунке 1.

Исходные коды программы распределены по соответствующим служебным директориям, включенным в главную директорию OpenFOAM decompose GUI. За запуск программы отвечает файл run.py. Он содержит инструкции визуализации главного окна и ряд базовых инструкций, описывающих логику работы приложения (рис. 2).

Результаты исследования

Итогом выполненной исследовательской работы является графическая оболочка OpenFOAM_decompose_GUI для проведения параллельных расчетов в OpenFOAM в рамках моделирования задач МСС в конструкторских и технологических подразделениях машиностроительных предприятий. Параллельные вычисления осуществляются в режиме синхронного выполнения на распределенных ресурсах. При этом сеточная модель и связанные поля разбиваются на отдельные компоненты и распределяются между процессорами (или ядрами процессора) для решения. По завершении параллельного запуска расчетный случай может быть реконструирован для последующей обработки, а также визуализации результатов на базе пакета ParaView. Параллельное вычисление обеспечивает экономию затрат времени на выполнение процесса численного моделирования задачи МСС.

Изображения главного окна программы (графического интерфейса пользователя), а также отдельных интерфейсных блоков представлены на рисунках (см. <http://www.swsys.ru/uploaded/image/2019-3/2019-3-dop/24.jpg>, <http://www.swsys.ru/uploaded/image/2019-3/2019-3-dop/25.jpg>, [```

graph LR
 Root\[OpenFOAM_decompose_GUI\] --- run\[run.py\]
 Root --- windows\[windows\]
 Root --- forms\[forms\]
 Root --- threads\[threads\]
 Root --- functions\[functions\]
 Root --- add_classes\[add_classes\]
 Root --- matches\[matches\]
 Root --- styles\[styles\]

 windows --- bMD\[bMD_window.py\]
 windows --- sHMD\[sHMD_window.py\]
 windows --- files_0\[files_0_window.py\]
 windows --- lng\[lng_window.py\]
 windows --- prj\[prj_window.py\]
 windows --- msh\[msh_window.py\]

 forms --- bMD_forms\[bMD_forms\]
 forms --- sHMD_forms\[sHMD_forms\]
 forms --- constant_forms\[constant_forms\]
 forms --- system_forms\[system_forms\]
 forms --- forms_0\[forms_0\]

 threads --- msh_threads\[msh_threads.py\]
 threads --- solv_threads\[solv_threads.py\]

 functions --- blockMeshDict\[blockMeshDict_generation.py\]
 functions --- snappyHexMeshDict\[snappyHexMeshDict_generation.py\]
 functions --- msh_functions\[msh_functions.py\]
 functions --- solv_functions\[solv_functions.py\]

 add_classes --- file_form_class\[file_form_class.py\]

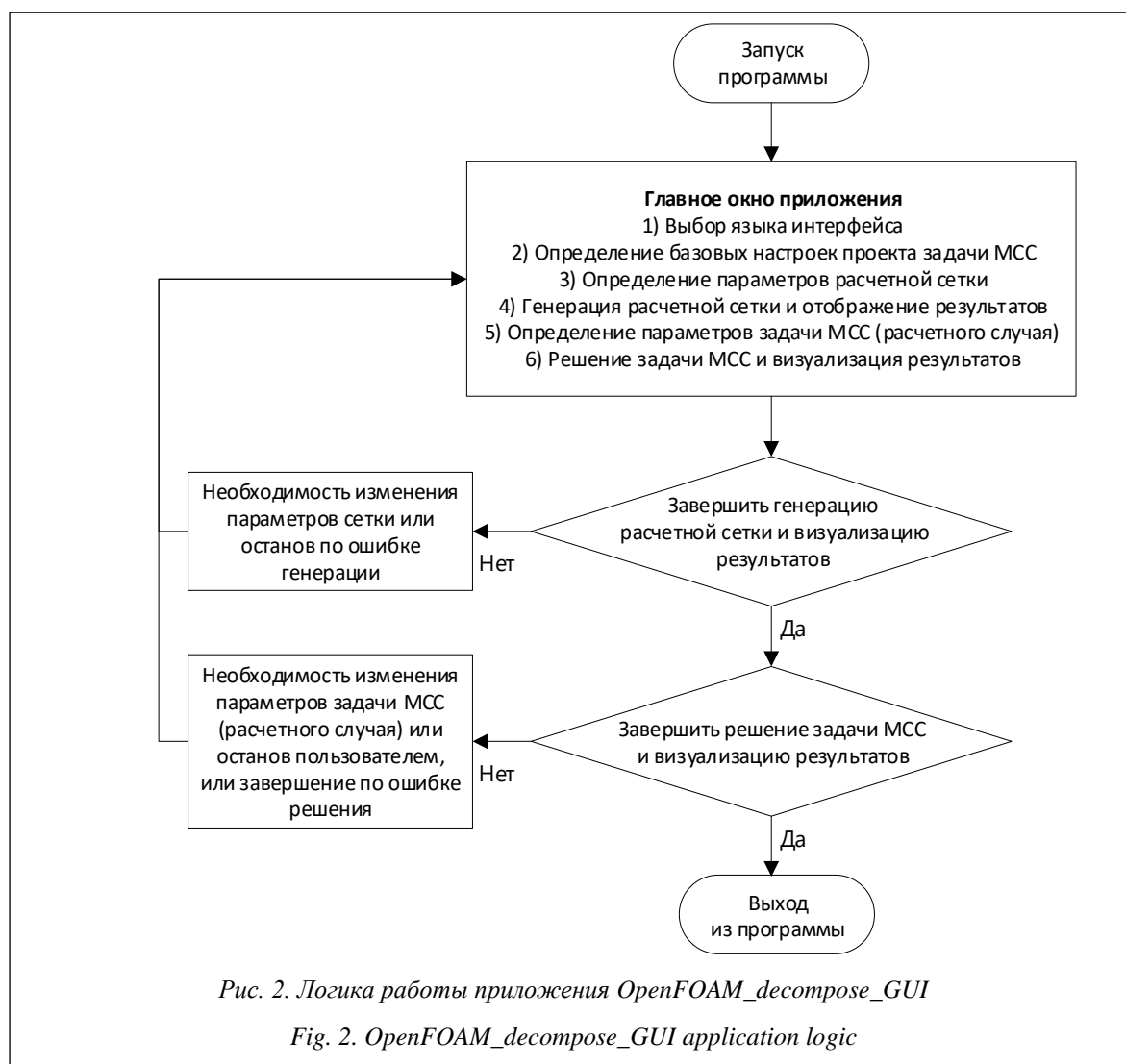
 matches --- Shablon\[Shablon\]
 matches --- blockMeshDict_matches\[blockMeshDict\]
 matches --- snappyHexMeshDict_matches\[snappyHexMeshDict\]

 styles --- properties_form_style\[properties_form_style.py\]

```](http://www.swsys.ru/up-</a></p>
</div>
<div data-bbox=)

Рис. 1. Структура приложения OpenFOAM\_decompose\_GUI

Fig. 1. OpenFOAM\_decompose\_GUI application structure



loaded/image/2019-3/2019-3-dop/26.jpg, <http://www.swsys.ru/uploaded/image/2019-3/2019-3-dop/27.jpg>, <http://www.swsys.ru/uploaded/image/2019-3/2019-3-dop/28.jpg>, <http://www.swsys.ru/uploaded/image/2019-3/2019-3-dop/29.jpg>). Тестирование приложения проведено на базе учебной задачи depthCharge3D, входящей в блок учебных примеров стандартного дистрибутива платформы OpenFOAM.

Предложенная программа OpenFOAM\_decompose\_GUI расширяет исходный код свободно распространяемой платформы OpenFOAM. Программа является универсальной и может быть внедрена на предприятиях различных отраслей машиностроения. Она находится в свободном доступе на платформе хостинга IT-проектов GitHub [21]. Приложение может быть протестировано потенциальными специалистами при решении учебных или реальных задач МСС на базе OpenFOAM.

## Заключение

Человеко-компьютерное взаимодействие – понятие, которое определяет аппаратные и программные средства, необходимые для обмена данными между человеком и вычислительным устройством. Современные интерфейсы являются неотъемлемым элементом любой программы, системы, платформы. Они используются и в офисной работе, и в процессе производственного цикла, когда специалисту необходимо передать вычислительному устройству некоторые данные и получить результат обработки, в частности, на этапе проектирования продукции.

Интерфейсы современных компьютерных программ, как правило, реализованы в виде графических элементов экрана, связывающих человека и компьютер. Оконные графические интерфейсы являются наиболее понятным

средством взаимодействия специалиста и вычислительного устройства. В настоящее время практически все коммерческое ПО включает программные интерфейсы пользователя. При этом коммерческое ПО требует приобретения лицензии, что может не вписываться в бюджет компании. Таким образом, предприятия переходят на открытые программные продукты, не уступающие по функциональным возможностям существующим коммерческим решениям. К таким программным продуктам относится платформа OpenFOAM, успешно зарекомендовавшая себя при моделировании задач МСС на предприятиях различных отраслей машиностроения.

Данная платформа функционирует на базе ОС Linux и традиционного графического интерфейса в своем дистрибутиве не имеет. Пользователи вынуждены вручную создавать проекты расчетных случаев, содержащие служебные директории и файлы параметров задач МСС, а также использовать командную строку для управления ходом численного моделирования. Это сложный и затратный по времени подход. В качестве альтернативы предлагается использование специализированных графических оболочек: Salome, Helyx-OS и Visual-CFD. Однако данный подход не лишен недостатков: не все продукты из названных находятся в свободном доступе, кроме того, предприятия вынуждены оплачивать работу специалистов по технической поддержке и консультантов. Таким образом, проблема отсутствия средства взаимодействия с пользователем для OpenFOAM сохраняет свою актуальность.

В настоящей работе представлена оригинальная графическая оболочка для проведения численных исследований на базе OpenFOAM, реализованная с помощью языка программирования Python 3.5 и фреймворка графических элементов PyQt5. Программа позволяет выполнять параллельные вычисления в рамках численного эксперимента, способствуя экономии ресурсов вычислительного устройства и времени проведения численного моделирования. Приложение находится в свободном доступе, обладает интуитивно понятным интерфейсом и не требует оплаты услуг по сопровождению. Программа размещена на сервисе GitHub и может применяться частными коллективами исследователей, а также на предприятиях маши-

ностроения, использующих в работе платформу OpenFOAM.

В программе реализованы механизмы, определяющие научную новизну разработки, прежде всего возможность подготовки любого количества расчетных сеток для одного расчетного случая (задачи МСС), возможность валидации вводимых данных и проверки комплектности расчетного случая, возможность создания служебных файлов с различными начальными условиями численного эксперимента, реализации программного запуска консольных команд OpenFOAM с помощью программной генерации и запуска средствами языка Python соответствующих bash-скриптов.

Использование созданной программы OpenFOAM\_decompose\_GUI позволяет заменить сложный и затратный по времени традиционный подход альтернативным, осуществлять все этапы численного эксперимента в OpenFOAM централизованно, посредством привычного оконного графического интерфейса. Этот подход дает ряд преимуществ.

Во-первых, экономию временных затрат. За создание расчетного случая, включая вложенные каталоги и служебные файлы с параметрами задачи МСС, отвечает программа, как и за запуск утилит препроцессинга, решения и постпроцессинга.

Во-вторых, минимизацию ошибок. Программа с помощью валидаторов проверяет корректность вводимых данных, а также комплектность файлов директорий расчетного случая перед запуском решения и визуализацией результатов.

В-третьих, упрощение освоения платформы OpenFOAM. Процесс численного моделирования с помощью программы OpenFOAM\_decompose\_GUI осуществляется по строго определенному алгоритму. Например, пользователь не может запустить расчетный случай на решение, не установив параметры сеточной модели.

Автор намерен продолжить работу в выбранном направлении и усовершенствовать текущую версию программы OpenFOAM\_decompose\_GUI, обеспечив специалистам доступ через графический интерфейс к другим утилитам платформы OpenFOAM в процессе выполнения препроцессинга, решения и постпроцессинга.

### Литература

1. Operating system interface design between 1981–2009. URL: <https://www.webdesignerdepot.com/2009/03/operating-system-interface-design-between-1981-2009/> (дата обращения: 25.01.2019).

2. OpenFOAM. The open source CFD toolbox. URL: <https://www.openfoam.com/> (дата обращения: 25.01.2019).
3. Etrati A., Frigaard I.A. Viscosity effects in density-stable miscible displacement flows: Experiments and simulations. *Physics of Fluids*, 2018, vol. 30, no. 12, art. 123104. DOI: 10.1063/1.5065388.
4. Salome. The open source integration platform for numerical simulation. URL: <http://www.salome-platform.org> (дата обращения: 25.01.2019).
5. Visual-CFD for OpenFOAM. URL: <https://www.esi-group.com/software-solutions/virtual-environment/cfd-multiphysics/visual-cfd-openfoam> (дата обращения: 25.01.2019).
6. HELYX-OS. The market leading open-source GUI for OpenFOAM. URL: <http://engys.com/products/helyx-os> (дата обращения: 25.01.2019).
7. Читалов Д.И., Меркулов Е.С., Калашников С.Т. Разработка графического интерфейса пользователя для программного комплекса OpenFOAM // Программная инженерия. 2016. № 12. С. 568–574. DOI: 10.17587/prin.7.568-574.
8. OpenFOAM. User Guide. URL: <http://foam.sourceforge.net/docs/Guides-a4/OpenFOAMUserGuide-A4.pdf> (дата обращения: 25.01.2019).
9. OpenFOAM. Tutorial Guide. URL: <https://www.openfoam.com/documentation/tutorial-guide/index.php> (дата обращения: 25.01.2019).
10. Galindo-Lopez S., Salehi F., Cleary M.J., Masri A.R., Neuber G., Stein O.T. A stochastic multiple mapping conditioning computational model in OpenFOAM for turbulent combustion. *Computers and Fluids*, 2018, vol. 172, pp. 410–425. DOI: 10.1016/j.compfluid.2018.03.083.
11. Kyriazis N., Koukouvinis P., Gavaises M. Modelling cavitation during drop impact on solid surfaces. *Advances in Colloid and Interface Science*, 2018, vol. 260, pp. 46–64. DOI: 10.1016/j.cis.2018.08.004.
12. Al-Jelawy H., Kaczmarczyk S., Alkhafaji D., Mirhadizadeh S., Lewis R., Cross M. A computational investigation of a turbulent flow over a backward facing step with OpenFOAM. *Proc. 9th Intern. Conf. DeSE, Liverpool*, 2016, pp. 301–307. DOI: 10.1109/DeSE.2016.47.
13. Mohseni M., Esperanca P.T., Sphaier S.H. Numerical study of wave run-up on a fixed and vertical surface-piercing cylinder subjected to regular, non-breaking waves using OpenFOAM. *APPL OCEAN RES*, 2018, vol. 79, pp. 228–252. DOI: 10.1016/j.apor.2018.08.003.
14. Читалов Д.И., Калашников С.Т. Разработка приложения для подготовки расчетных сеток с градуируемыми и изогнутыми краями для программной среды OpenFOAM // Системы и средства информатики. 2018. Т. 28. № 4. С. 122–135. DOI: 10.14357/08696527180412.
15. Читалов Д.И., Калашников С.Т. Разработка приложения для подготовки расчетных сеток посредством утилиты snappyHexMesh программной среды OpenFOAM // Программные продукты и системы. 2018. Т. 31. № 4. С. 715–722. DOI: 10.15827/0236-235X.124.715-722.
16. Advanced bash-scripting guide. URL: <https://www.tldp.org/LDP/abs/html/> (дата обращения: 25.01.2019).
17. TIOBE index for January 2019. URL: <https://www.tiobe.com/tiobe-index/> (дата обращения: 25.01.2019).
18. Python 3.5 Documentation. URL: <https://docs.python.org/3.5/> (дата обращения: 25.01.2019).
19. PyQt5 Reference Guide. URL: <http://pyqt.sourceforge.net/Docs/PyQt5/> (дата обращения: 25.01.2019).
20. ParaView. URL: <https://www.paraview.org/> (дата обращения: 25.01.2019).
21. OpenFOAM\_decompose\_GUI. URL: [https://github.com/DmitryChitalov/OpenFOAM\\_decompose\\_GUI](https://github.com/DmitryChitalov/OpenFOAM_decompose_GUI) (дата обращения: 25.01.2019).

## Graphical shell development for parallel calculations based on the OpenFOAM platform

*D.I. Chitalov*<sup>1</sup>, Junior Researcher, [cdi9@yandex.ru](mailto:cdi9@yandex.ru)

<sup>1</sup> South Ural Federal Research Centre of Mineralogy and Geoecology of the UB RAS, Miass, Ilmen reserve, 456317, Russian Federation

**Abstract.** Graphical interfaces are an essential element of human-computer interaction. They form input data for programs and visualize results. Commercial software mainly comes with built-in graphical interaction

tools. At the same time, some open software solutions (the OpenFOAM platform in particular) do not have built-in interaction tools. This problem remains relevant since the existing graphical interfaces have disadvantages.

This paper presents a step-by-step process of developing a graphical shell for implementing user interaction with the OpenFOAM platform, i.e. for setting up numerical experiments applied to problems of continuum mechanics in the parallel execution mode. To achieve this goal, there is a defined list of tasks and necessary tools: Python 3.5 programming language, PyQt5 interface element description framework, PyCharm integrated development environment.

The paper gives diagrams showing the interconnection of program modules and a program operation mechanism. The results of the application development and testing are formed by the example of a project of one of the standard tasks of continuum mechanics, which are a part of the OpenFOAM platform distribution kit. The authors provide scientific novelty of the research, in particular: serialization of numerical experiment parameters using the Python Pickle module and database tables in the SQLite format; the ability to create different versions of files with continuum mechanics task parameters; the ability to run console commands using bash scripts. They also identify the practical value of the work and future prospects.

**Keywords:** numerical simulation, continuum mechanics, parallel calculations, OpenFOAM, open source software, graphical user interface, Python programming language, PyQt5 library, SQLite DBMS, bash-scripting.

### References

1. *Operating System Interface Design Between 1981–2009*. Available at: <https://www.webdesignerdepot.com/2009/03/operating-system-interface-design-between-1981-2009/> (accessed January 25, 2019).
2. *OpenFOAM. The Open Source CFD Toolbox*. Available at: <https://www.openfoam.com/> (accessed January 25, 2019).
3. Etrati A., Frigaard I.A. Viscosity effects in density-stable miscible displacement flows: Experiments and simulations. *Physics of Fluids*. 2018, vol. 30, no. 12. DOI: 10.1063/1.5065388.
4. *Salome. The Open Source Integration Platform for Numerical Simulation*. Available at: <http://www.salome-platform.org> (accessed January 25, 2019).
5. *Visual-CFD for OpenFOAM*. Available at: <https://www.esi-group.com/software-solutions/virtual-environment/cfd-multiphysics/visual-cfd-openfoam> (accessed January 25, 2019).
6. *HELYX-OS. The Market Leading Open-Source GUI for OpenFOAM*. Available at: <http://engys.com/products/helyx-os> (accessed January 25, 2019).
7. Chitalov D.I., Merkulov E.S., Kalashnikov S.T. Development of a graphical user interface for the OpenFOAM software package. *Software Engineering*. 2016, iss. 12, pp. 568–574. DOI: 10.17587/prin.7.568-574 (in Russ.).
8. *OpenFOAM. User Guide*. Available at: <http://foam.sourceforge.net/docs/Guides-a4/OpenFOAMUserGuide-A4.pdf> (accessed January 25, 2019).
9. *OpenFOAM. Tutorial Guide*. Available at: <https://www.openfoam.com/documentation/tutorial-guide/index.php> (accessed January 25, 2019).
10. Galindo-Lopez S., Salehi F., Cleary M.J., Masri A.R., Neuber G., Stein O.T. A stochastic multiple mapping conditioning computational model in OpenFOAM for turbulent combustion. *Computers and Fluids*. 2018, iss. 172, pp. 410–425. DOI: 10.1016/j.compfluid.2018.03.083.
11. Kyriazis N., Koukouvinis P., Gavaises M. Modelling cavitation during drop impact on solid surfaces. *Advances in Colloid and Interface Science*. 2018, iss. 260, pp. 46–64. DOI: 10.1016/j.cis.2018.08.004.
12. Al-Jelawy H., Kaczmarczyk S., AlKhafaji D., Mirhadizadeh S., Lewis R., Cross M. A computational investigation of a turbulent flow over a backward facing step with OpenFOAM. *Proc. 9th Intern. Conf. on Developments in eSystems Engineering (DeSE)*. Liverpool, 2016, pp. 301–307. DOI: 10.1109/DeSE.2016.47.
13. Mohseni M., Esperanca P.T., Sphaier S.H. 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*. 2018, iss. 79, pp. 228–252. DOI: 10.1016/j.apor.2018.08.003.
14. Chitalov D.I., Kalashnikov S.T. Development of an application for the preparation of computational grids with graduating and curved edges for the OpenFOAM software environment. *Systems and Means of Informatics*. 2018, vol. 28, iss. 4, pp. 122–135. DOI: 10.14357/08696527180412 (in Russ.).
15. Chitalov D.I., Kalashnikov S.T. Developing an application for preparing calculation grids using the snappyHexMesh utility of the OpenFOAM software environment. *Software & Systems*. 2018, vol. 31, iss. 4, pp. 715–722. DOI: 10.15827/0236-235X.124.715-722.
16. *Advanced Bash-Scripting Guide*. Available at: <https://www.tldp.org/LDP/abs/html/> (accessed January 25, 2019).

17. *TIOBE Index for January 2019*. Available at: <https://www.tiobe.com/tiobe-index/> (accessed January 25, 2019).
18. *Python 3.5 Documentation*. Available at: <https://docs.python.org/3.5/> (accessed January 25, 2019).
19. *PyQt5 Reference Guide*. Available at: <http://pyqt.sourceforge.net/Docs/PyQt5/> (accessed January 25, 2019).
20. *ParaView*. Available at: <https://www.paraview.org/> (accessed January 25, 2019).
21. *OpenFOAM\_decompose\_GUI*. Available at: [https://github.com/DmitryChitalov/OpenFOAM\\_decompose\\_GUI](https://github.com/DmitryChitalov/OpenFOAM_decompose_GUI) (accessed January 25, 2019).

#### Для цитирования

Читалов Д.И. Разработка графической оболочки для параллельных расчетов на базе платформы OpenFOAM // Программные продукты и системы. 2019. Т. 32. № 3. С. 433–443. DOI: 10.15827/0236-235X.127.433-443.

#### For citation

Chitalov D.I. Graphical shell development for parallel calculations based on the OpenFOAM platform. *Software & Systems*. 2019, vol. 32, no. 3, pp. 433–443 (in Russ.). DOI: 10.15827/0236-235X.127.433-443.