

Электронный научный журнал "Математическое моделирование, компьютерный и натурный эксперимент в естественных науках" <http://mathmod.esrae.ru/>

URL статьи: [mathmod.esrae.ru/41-161](http://mathmod.esrae.ru/41-161)

Ссылка для цитирования этой статьи:

Панкратов И.А., Симонова К.Р. О реализации информационной системы пре-процессинга данных для численных экспериментов // Математическое моделирование, компьютерный и натурный эксперимент в естественных науках. 2023. № 1

УДК 519.6, 004.65

DOI: 10.24412/2541-9269-2023-1-06-18

## О РЕАЛИЗАЦИИ ИНФОРМАЦИОННОЙ СИСТЕМЫ ПРЕ-ПРОЦЕССИНГА ДАННЫХ ДЛЯ ЧИСЛЕННЫХ ЭКСПЕРИМЕНТОВ

Панкратов И.А.<sup>1</sup>, Симонова К.Р.<sup>2</sup>

<sup>1</sup>Саратовский национальный исследовательский государственный университет им. Н.Г. Чернышевского, Россия, Саратов, [pankratovia@info.sgu.ru](mailto:pankratovia@info.sgu.ru)

<sup>2</sup>Саратовский национальный исследовательский государственный университет им. Н.Г. Чернышевского, Россия, Саратов, [simonovakr@info.sgu.ru](mailto:simonovakr@info.sgu.ru)

## ON THE IMPLEMENTATION OF INFORMATION SYSTEM OF DATA PRE-PROCESSING FOR NUMERICAL EXPERIMENTS

Pankratov I.A.<sup>1</sup>, Simonova K.R.<sup>2</sup>

<sup>1</sup>Saratov State University, Russia, Saratov, [pankratovia@info.sgu.ru](mailto:pankratovia@info.sgu.ru)

<sup>2</sup>Saratov State University, Russia, Saratov, [simonovakr@info.sgu.ru](mailto:simonovakr@info.sgu.ru)

**Аннотация.** В работе построена информационная система для проведения расчётов в пакете OpenFOAM. Для хранения данных использован нереляционный подход. Разработанное графическое приложение позволяет частично автоматизировать проведение численного моделирования физических процессов в OpenFOAM.

Ключевые слова: OpenFOAM, графическое приложение, база данных, автоматизация, NoSQL.

**Abstract.** The information system for performing calculations in OpenFOAM is built in the paper. A non-relational approach was used for data storage. The graphical application developed in the paper allows partially automate the numerical simulation of physical processes in OpenFOAM.

Keywords: OpenFOAM, graphical application, database, automation, NoSQL.

В настоящей работе спроектирована и реализована информационная система (ИС) для того, чтобы частично автоматизировать проведение математического моделирования различных физических процессов в пакете OpenFOAM [1, 2]. Существующий подход к использованию платформы OpenFOAM предусматривает использование только командной строки и требует от специалиста знания команд, предназначенных для запуска утилит OpenFOAM, необходимых для управления численными исследованиями, а

также структуры каждого расчётного случая (конкретной задачи механики сплошных сред (МСС)) и свойств задач МСС в различных служебных файлах. При существующем подходе исследователь вынужден работать с расчетным случаем через командную строку и вручную редактировать служебные файлы. Указанный подход весьма неудобен и требует существенных затрат рабочего времени. Альтернативой данному подходу является разработка и использование классического оконного графического интерфейса, благодаря которому процесс численного эксперимента на основе платформы OpenFOAM будет осуществляться с применением окон и элементов управления: кнопок, выпадающих списков, текстовых полей и т. д. Актуальность проблемы подтверждается интересом к ней отечественных и зарубежных разработчиков. Ряд иностранных компаний запустили собственные проекты по созданию полноценных графических оболочек для OpenFOAM. Их результатом стали программные решения такие, как Helyx-OS, Salome и Visual-Environment [3-5], которые обеспечивают выполнение всех этапов процесса численного моделирования. Создание вышеперечисленных программных продуктов для взаимодействия с OpenFOAM вызвало неоднозначную реакцию среди участников международного сообщества исследователей. С одной стороны, пользователи получили возможность отказаться от трудоёмкого традиционного подхода проведения численного моделирования, а с другой – в каждой из предложенных графических оболочек были обнаружены недостатки, из-за которых не все пользователи оказались готовы к переходу от традиционного подхода к использованию оконного интерфейса. Таким образом, у исследователей сохраняется интерес к вопросу создания графической оболочки для платформы OpenFOAM [6-8].

В настоящей работе рассмотрена задача автоматизации предварительной обработки данных для численных экспериментов, т.е. автоматизация работы препроцессора. Основные задачи препроцессора – задание геометрии расчетной области, построение сетки и постановка граничных условий. В состав OpenFOAM входят препроцессоры blockMesh и snappyHexMesh. Первый наиболее подходит для случая простой геометрии области. Утилита blockMesh позволяет строить структурированные сетки, состоящие из гексаэдров. Основным принцип работы blockMesh – деление всей расчетной области (заметим, что в OpenFOAM все задачи изначально формулируются в трехмерной постановке) на трехмерные гексаэдральные блоки. Ребрами этих блоков могут быть отрезки прямых либо дуги. Сетка получается разбиением каждого блока на заданное число ячеек в направлении каждой из осей прямоугольной декартовой системы координат. При этом ячейки могут быть разного размера. Вся необходимая для этого информация указывается в одном текстовом файле. Утилита snappyHexMesh предоставляет возможности для построения сеток в случае более сложной геометрии расчетной области. При решении реальных задач механики сплошных сред геометрия области, как правило, очень сложная. В таких случаях можно построить модель в CAD-

системах (например, AutoCAD, OpenSCAD) и экспортировать ее в формате STL (Stereolithography) [9], который поддерживается утилитой snappyHexMesh. Основной принцип ее работы – построение грубой сетки во всей области и последовательное ее уточнение с помощью разбиения ячеек, пересекающихся с границами расположенных в ней областей. Процесс продолжается до тех пор, пока не будет достигнут нужный размер сетки. После этого вершины ячеек сетки, пересекающих границы областей, переносятся на эти границы. Для повышения разрешающей способности snappyHexMesh может добавить несколько слоев структурированной сетки вблизи границы.

Вне зависимости от использованных для построения сетки средств, по завершении ее генерации требуется задать условия на границе расчетной области. Задание граничных условий выполняется в два этапа. Сначала для каждого отдельного участка границы указывается базовый тип граничного условия. Этот тип определяется исключительно геометрией участка или обменом данными между процессорами (при декомпозиции области для проведения параллельных расчетов). К базовым типам относятся, в частности, patch (участок общего вида), symmetryPlane (плоскость симметрии), wedge (треугольная призма – элемент геометрии осесимметричной задачи), empty (границы области, перпендикулярно которым решение задачи не меняется; этот тип появляется при переходе к решению двумерных или одномерных задач), processor (участок границы между областями, отнесенными к двум разным процессорам).

После указания базового типа участка границы задается конкретный вид граничного условия. Он определяется используемой математической моделью. Вид граничного условия может быть простым или производным. Простых типов достаточно мало. Среди них, в частности, типы, реализующие граничные условия I или II рода. Производных типов в OpenFOAM значительно больше. К ним относится, например, условие прилипания slip. Для выполнения расчета, помимо граничных условий, необходимо задать и начальные условия.

Модель ИС была создана с помощью унифицированного языка моделирования UML [10, 11]. На рис. 1 представлена диаграмма последовательностей для утилиты blockMesh. На данной диаграмме показаны аспекты подготовки расчетной области с помощью blockMesh.

Также на рис. 2 представлена диаграмма последовательностей для утилиты snappyHexMesh. На данной диаграмме показаны аспекты подготовки расчетной области с помощью snappyHexMesh.

Разработанное приложение OpenFOAM\_PA\_tool реализовано в виде набора модулей, который формирует общую структуру программы. Исходные коды программы распределены по соответствующим поддиректориям внутри главной директории OpenFOAM\_PA\_tool. За запуск программы и отображение стартового окна приложения отвечает файл \_\_main\_\_.py.

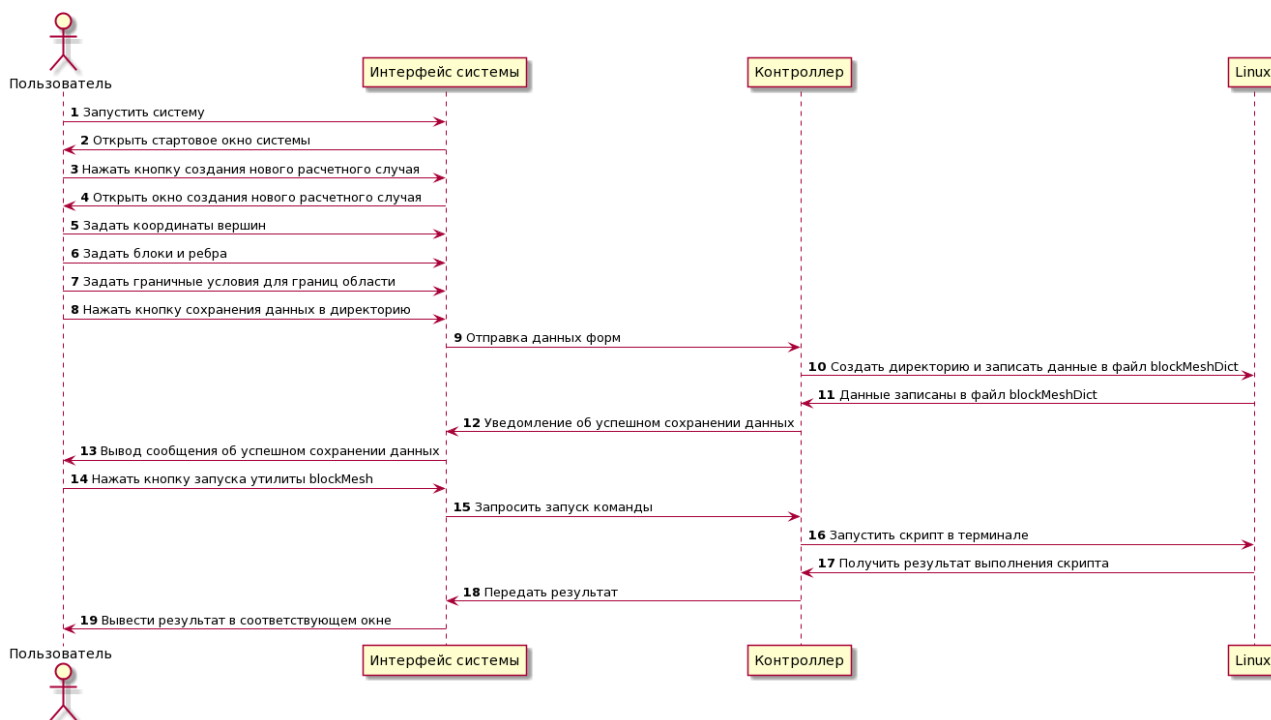


Рис. 1. Диаграмма последовательностей для утилиты blockMesh

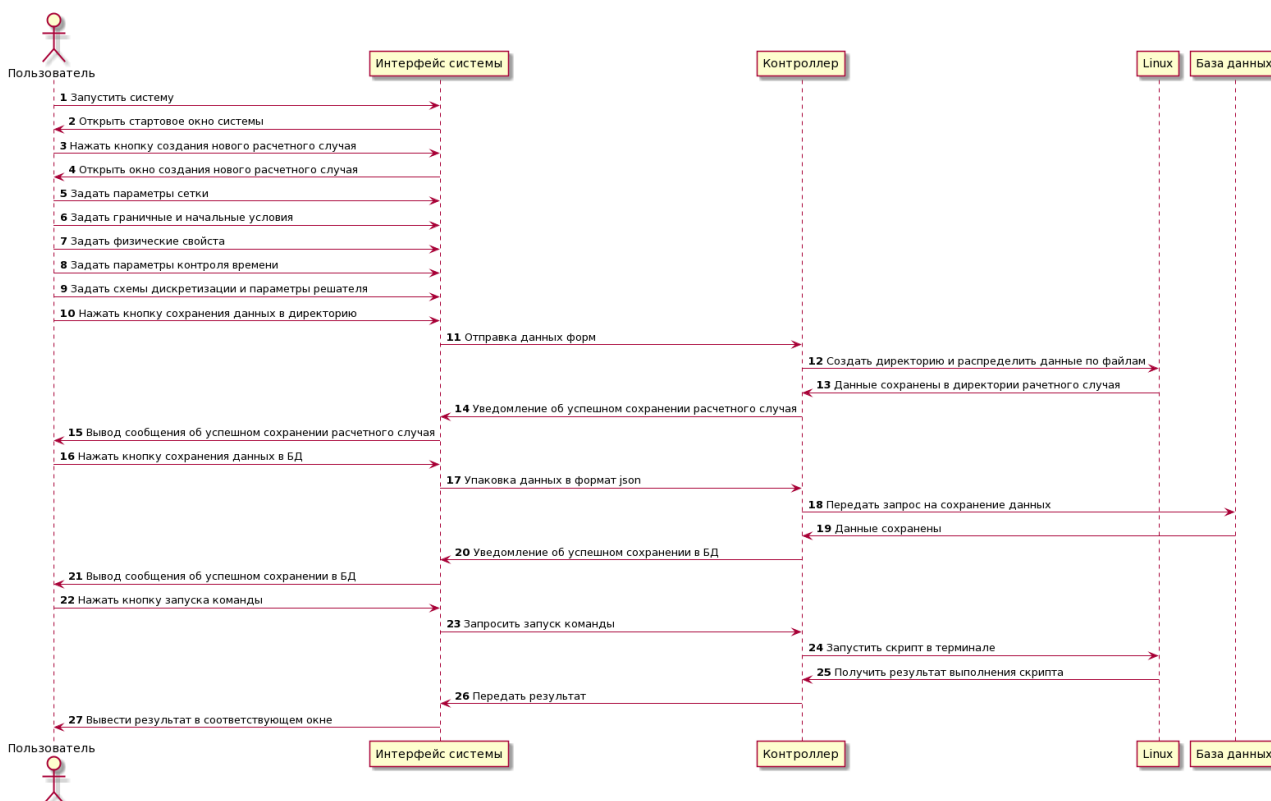


Рис. 2. Диаграмма последовательностей для утилиты snappyHexMesh

Директория `./windows/` содержит шесть служебных файлов. Первый из них, `start_window.py`, содержит программный код, отвечающий за отображение и логику работы стартового окна при создании нового расчетного случая или

при открытии расчетного случая, сохраненного ранее в базе данных (БД). Второй, `main_window.py`, обеспечивает функционирование главного окна программы. Диалоговое окно `new_case_dwin.py` отвечает за получение параметров создания нового расчетного случая от пользователя. Диалоговое окно `open_case_dwin.py` отвечает за получение имени расчетного случая, который необходимо открыть, из списка случаев, сохраненных ранее в БД. Следующие три файла директории: `bMD_window.py`, `sHMD_window.py` и `system_window.py` обеспечивают отображение набора форм, каждая из которых соответствует определенному блоку параметров расчетного случая. Программный код каждой из экранных форм хранится в отдельном файле, что упрощает дальнейшую модификацию форм.

Поскольку описываемая в настоящей работе программа предусматривает работу с утилитами `blockMesh` и `snappyHexMesh`, для этих утилит подготовлены две отдельные директории с наборами экранных форм `../bMD_forms/` и `../sHMD_forms/`. Помимо этого директория `../bMD_forms/` содержит форму для указания начальных условий (каталог `../0/` в `OpenFOAM`). Директория `../system_forms/` содержит формы для файлов из каталогов `../system/` и `../constant/` расчетного случая в `OpenFOAM`. Соответствующие формы импортируются в служебные файлы `bMD_window.py`, `sHMD_window.py` и `system_window.py`, которые отвечают за их объединение в набор форм и отображение его в блоке подготовки исходных данных главного окна программы.

Директория `../libs/` содержит классы `case_saver.py`, `caseDB_saver.py` и `caseDB_loader.py`. В первом классе реализованы функции, обеспечивающие считывание данных с форм; генерацию файлов, содержащих параметры расчетного случая, и их распределение по каталогам директории расчетного случая. Указанная директория создается при первом сохранении расчетного случая; при последующих сохранениях осуществляется перезапись его служебных файлов. Во втором – функции записи расчетного случая в базу данных `OpenFOAM_cases`. В третьем – функции получения данных из БД и их распределения по элементам форм.

Четвертая вложенная директория, `../bin/`, содержит `bash`-скрипты, отвечающие за запуск процессов генерации сеток, визуализации результатов, прерывания работы команд, а также очистки расчетного случая перед проведением новых вычислений. Помимо этого в данной директории расположены скрипты, отвечающие за проверку существования директории расчетного случая и его служебных файлов.

За инициализацию ключевых свойств сетки отвечает отдельное диалоговое окно. На основе внесенных в него значений базовых параметров автоматически генерируются параметры в остальных формах, тем самым образуя общую структуру файлов `blockMeshDict` и `snappyHexMeshDict`. Логика функционирования форм реализуется частично в служебных файлах `bMD_window.py`, `sHMD_window.py` и `system_window.py`, в которые

импортируются файлы экранных форм, и частично в самих файлах форм. Для реализации программного запуска консольных команд OpenFOAM используется bash-скриптинг [12, 13].



```
Terminal File Edit View Search Terminal Help
> use OpenFOAM_cases
switched to db OpenFOAM_cases
> db.getCollection('/home/karina/Desktop/example/').find().pretty()
{
  "_id" : "blockMeshDict",
  "mergePatchPairs" : [ ],
  "boundary" : [
    {
      "Toppatch1" : [
        "empty",
        "(7 6 2 3)"
      ],
      "Rightpatch1" : [
        "wall",
        "(5 1 2 6)"
      ],
      "Bottompatch1" : [
        "empty",
        "(4 5 1 0)"
      ],
      "Frontpatch1" : [
        "empty",
        "(4 5 6 7)"
      ],
      "Backpatch1" : [
        "empty",
        "(0 1 2 3)"
      ],
      "Leftpatch1" : [
        "wall",
        "(4 0 3 7)"
      ]
    }
  ],
  "blocks" : [
    [
      "(0 1 2 3 4 5 6 7)",
      "(20 20 20)",
      "(1 1 1)"
    ]
  ],
  "date" : ISODate("2020-05-18T15:39:54.982Z"),
  "convertToMeters" : "1",
  "edges" : [
    {
    }
  ],
  "vertices" : [
    [
      "(0.0 0.0 0.0)",
      "(0.0 1.0 0.0)",
      "(0.0 1.0 1.0)",
      "(0.0 0.0 1.0)",
      "(1.0 0.0 0.0)",
      "(1.0 1.0 0.0)",
      "(1.0 1.0 1.0)",
      "(1.0 0.0 1.0)"
    ]
  ]
}
```

Рис. 3. Структура документа в коллекции

Результаты работы ИС сохраняются в БД OpenFOAM\_cases. В ней находятся коллекции, названия которых соответствуют названиям



расчетных случаев. Эти коллекции содержат в себе документы, каждый из которых является аналогом файла из директории расчетного случая в OpenFOAM. Для хранения данных был выбран NoSQL подход [14-16], а в качестве СУБД – MongoDB [17, 18]. Заметим, что NoSQL-подход часто применяется и другими авторами при анализе плохо структурированных данных (см., например, работы [19-22]). В зависимости от расчетного случая документы в коллекциях имеют разную структуру, исходя их специфики решения задачи: области применения, модели сплошной среды, геометрии расчетной области, граничных и начальных условий и так далее. В таких условиях сложности структур документов документо-ориентированные БД имеют огромное преимущество по сравнению с реляционными. Структура документа коллекции в формате JSON, соответствующего файлу blockMeshDict, показана на рис. 3.

Таким образом, в процессе разработки программного продукта использовались следующие подходы:

1. Создание набора форм для редактирования параметров файлов, соответствующих расчетному случаю. Например, в файлах blockMeshDict и snappyHexMeshDict 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. Так упрощается подготовка сетки конечным пользователем.

2. Создание БД для файлов параметров расчетного случая. Для каждого из расчетных случаев создается коллекция в MongoDB, где каждый документ 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.

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

Далее приведен пример работы разработанной программы. После выбора опции создания нового расчетного случая в стартовом окне приложения управление передается на главное окно. Так как предполагается совместное использование утилит snappyHexMesh и blockMesh, то по нажатию соответствующей кнопки добавляются формы для указания параметров файла snappyHexMeshDict.

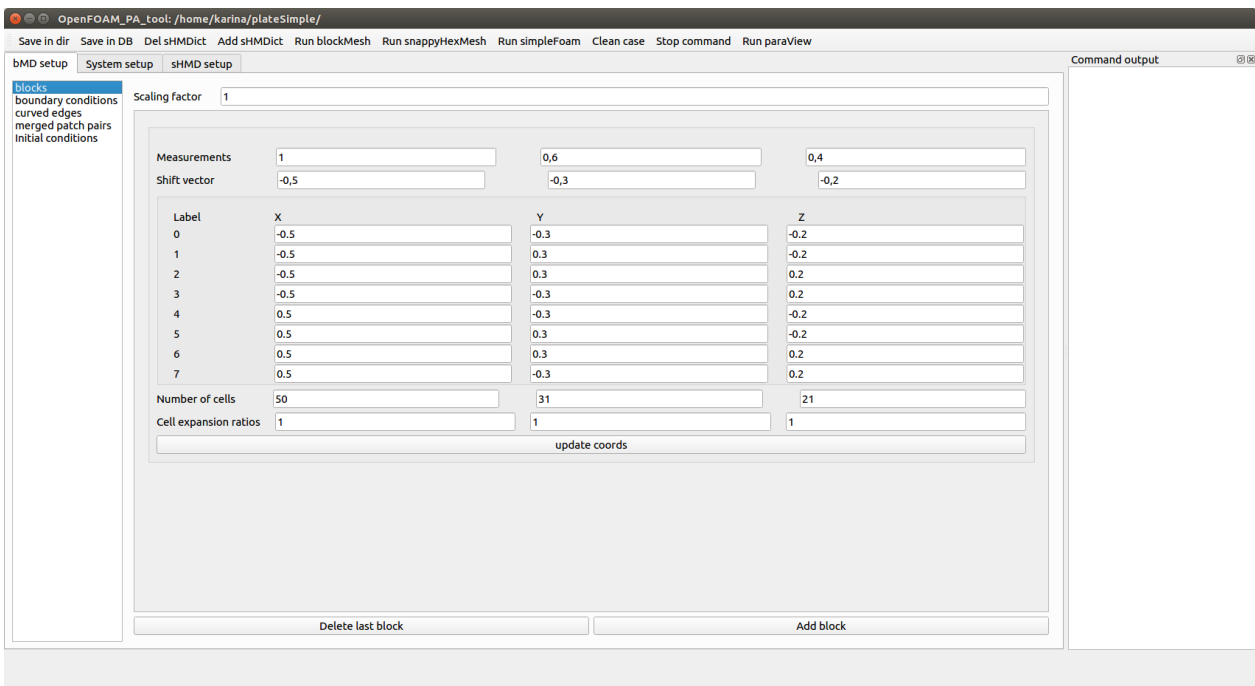


Рис. 4. Форма «blocks»

В главном окне в форме «blocks» вкладки «bMD setup» на основе базовых параметров, таких как размерность, вектор сдвига относительно начала координат, число ячеек в каждом направлении и коэффициенты расширения ячеек, генерируются координаты вершин с метками для каждой вершины блока, что показано на рис. 4.

Для данного блока в остальных формах создаются соответствующие параметры. В форме «boundary conditions» генерируются области с названиями по умолчанию и для каждой из них указываются граничные условия, что показано на рис. 5.

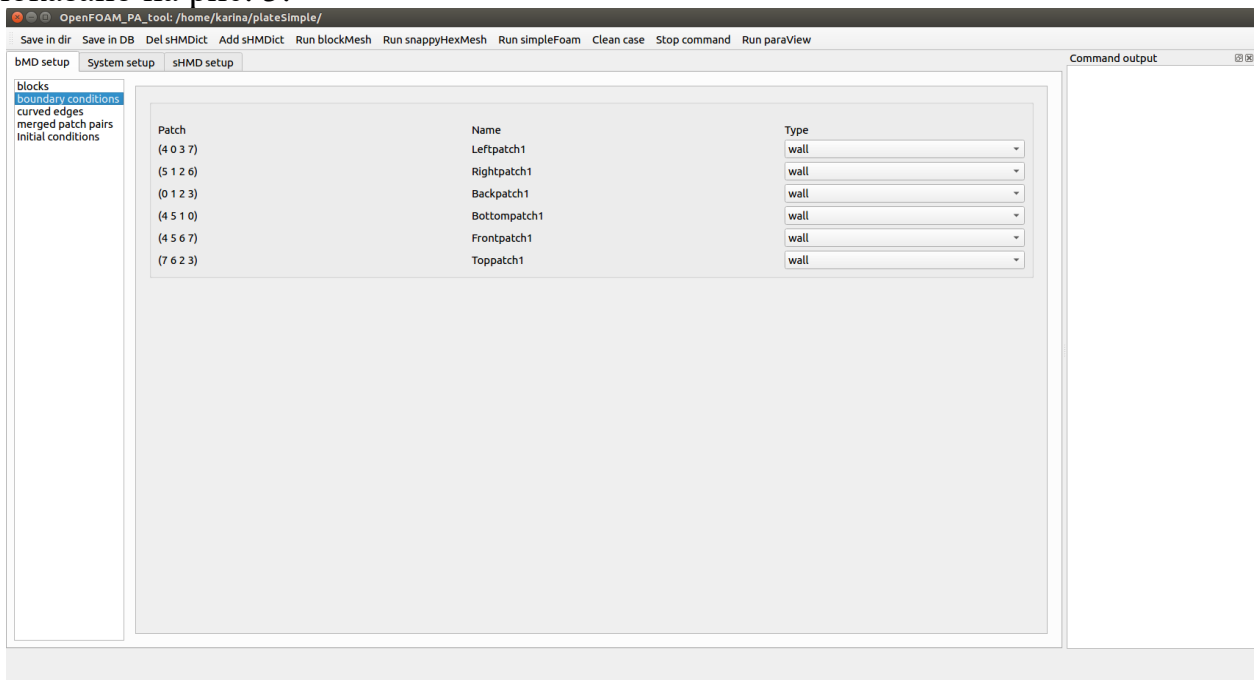


Рис. 5. Форма «boundary conditions»



В форму «geometry» вкладки «SHMD setup» загружаются stl-объекты посредством нажатия кнопки выбора требуемого файла, при этом поле «file» принимает название файла вместе с его расширением, а сами данные считываются из файла и хранятся во внутренней переменной. Также в данной форме указываются регионы улучшения для stl-поверхностей, что показано на рис. 6.

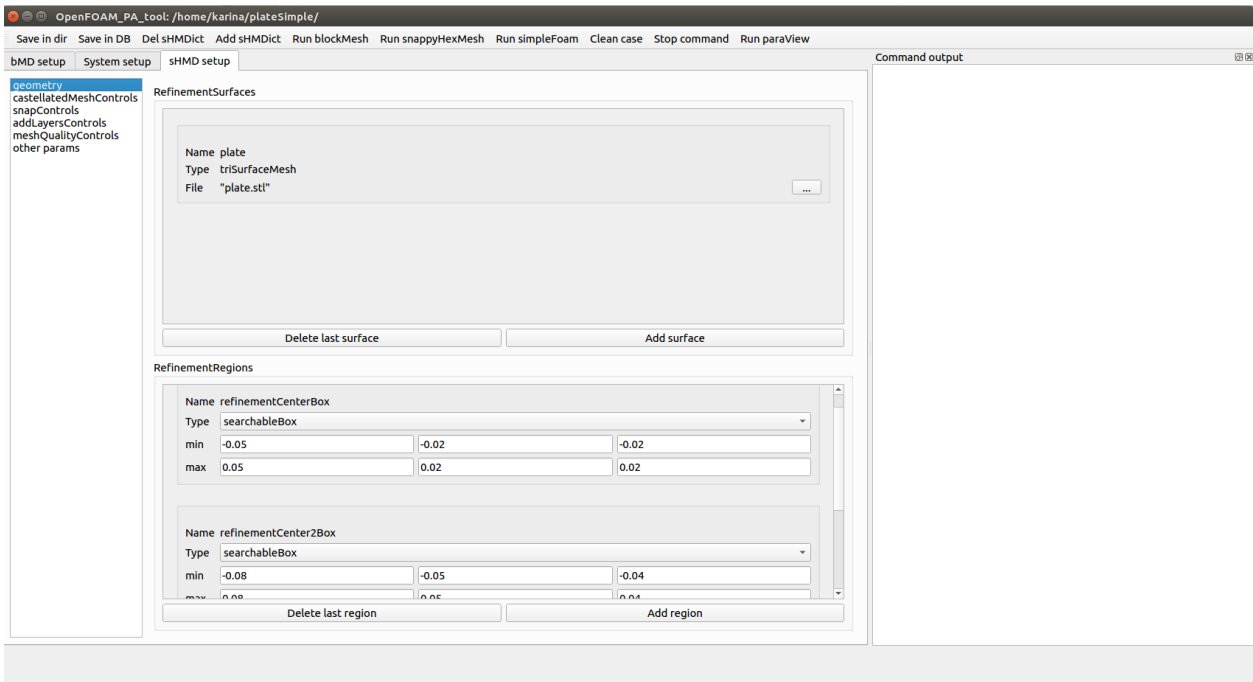


Рис. 6. Форма «geometry»

В форме «castellatedMeshControls» для поверхностей и регионов задаются уровни улучшения и другие параметры, что показано на рис. 7. Остальные формы данной вкладки заполняются аналогично.

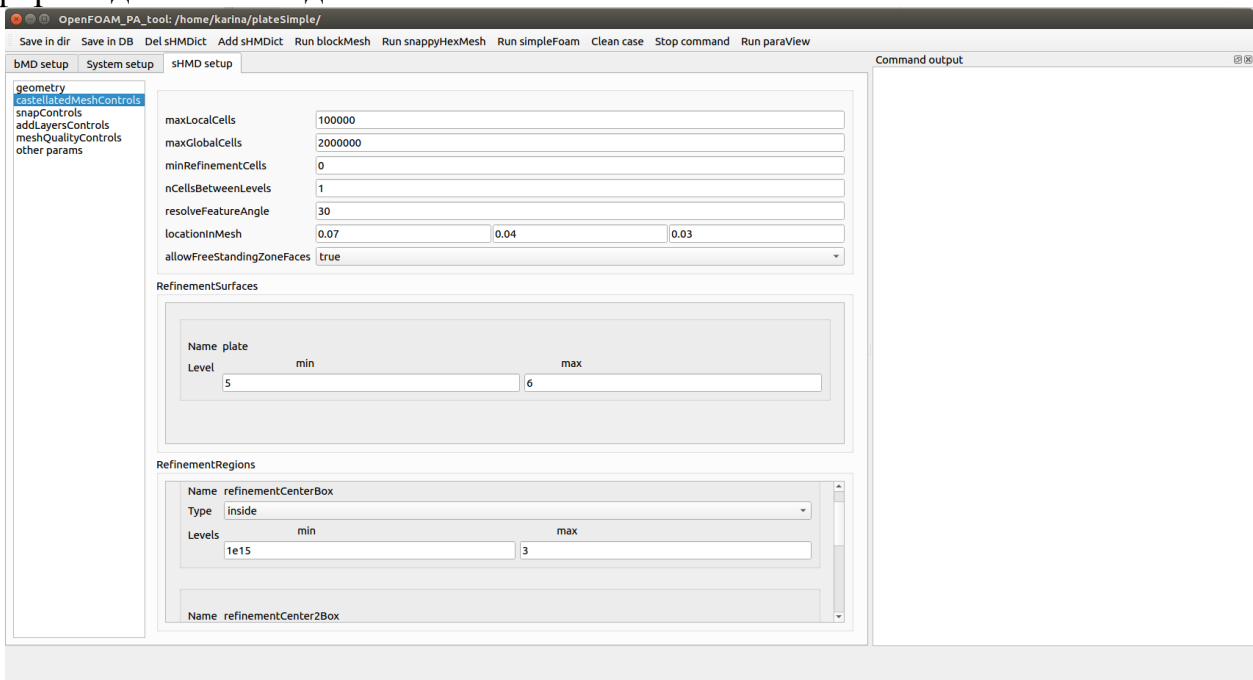


Рис. 7. Форма «castellatedMeshControls»

Далее в форме «Initial conditions» задаются начальные условия для скорости и давления, а именно значения этих величин внутри и на границах областей. Так как при задании параметров для snappyHexMeshDict была добавлена stl-поверхность, то она автоматически появляется в данной форме и для нее также указываются начальные условия, что показано на рис. 8.

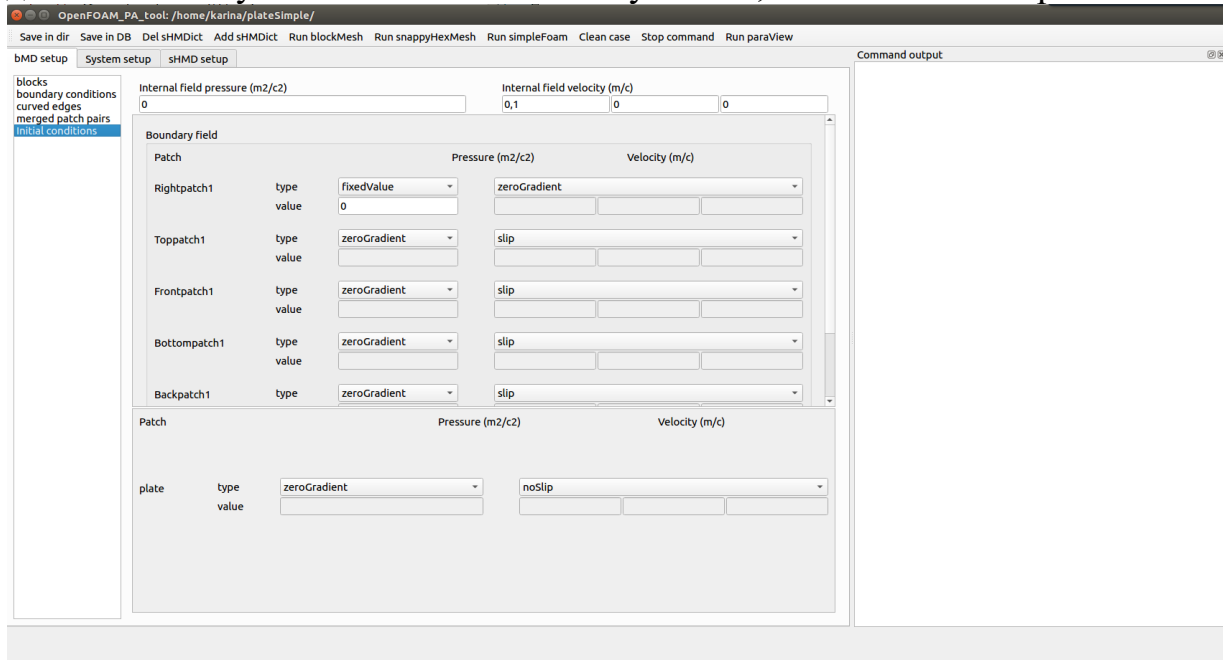


Рис. 8. Форма «Initial conditions»

Параметры форм вкладки «system setup» являются базовыми для решателя simpleFoam (расчёт параметров течения вязкой несжимаемой жидкости в стационарном случае), они остаются по умолчанию. Когда все исходные данные заданы, нажатием по соответствующей кнопке запускается команда построения сетки blockMesh, что показано на рис. 9.

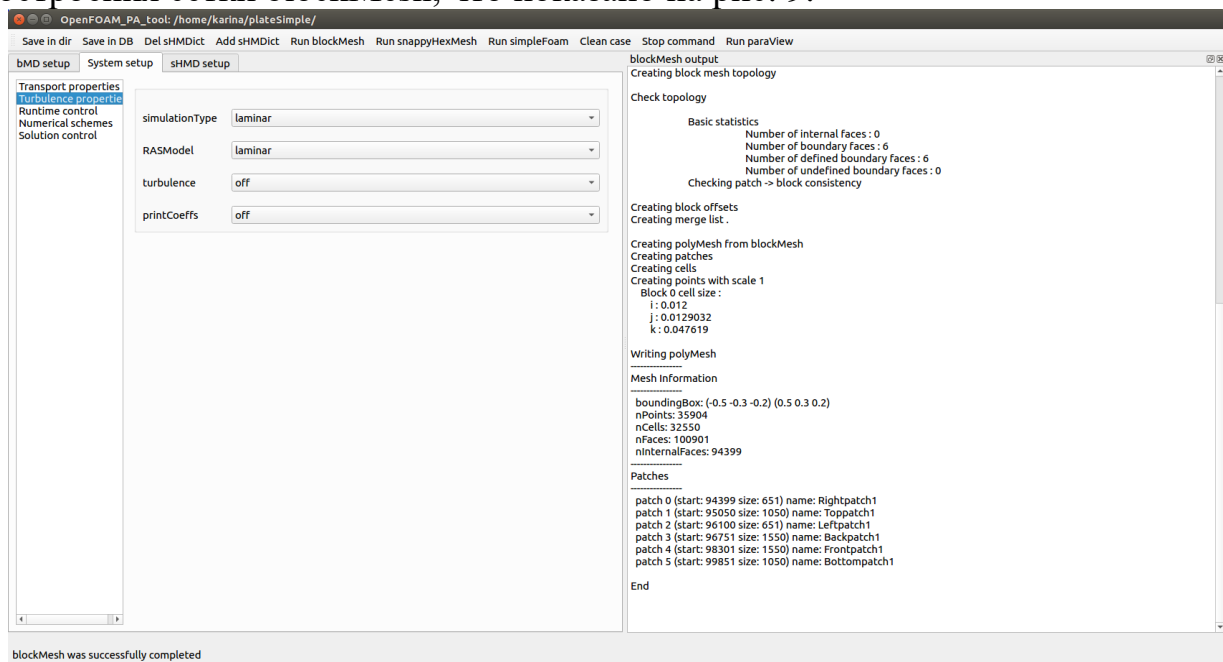


Рис. 9. Результат работы команды blockMesh

Команда blockMesh отработала без ошибок (блочная сетка построена успешно), поэтому запускается на выполнение команда snappyHexMesh, что показано на рис. 10.

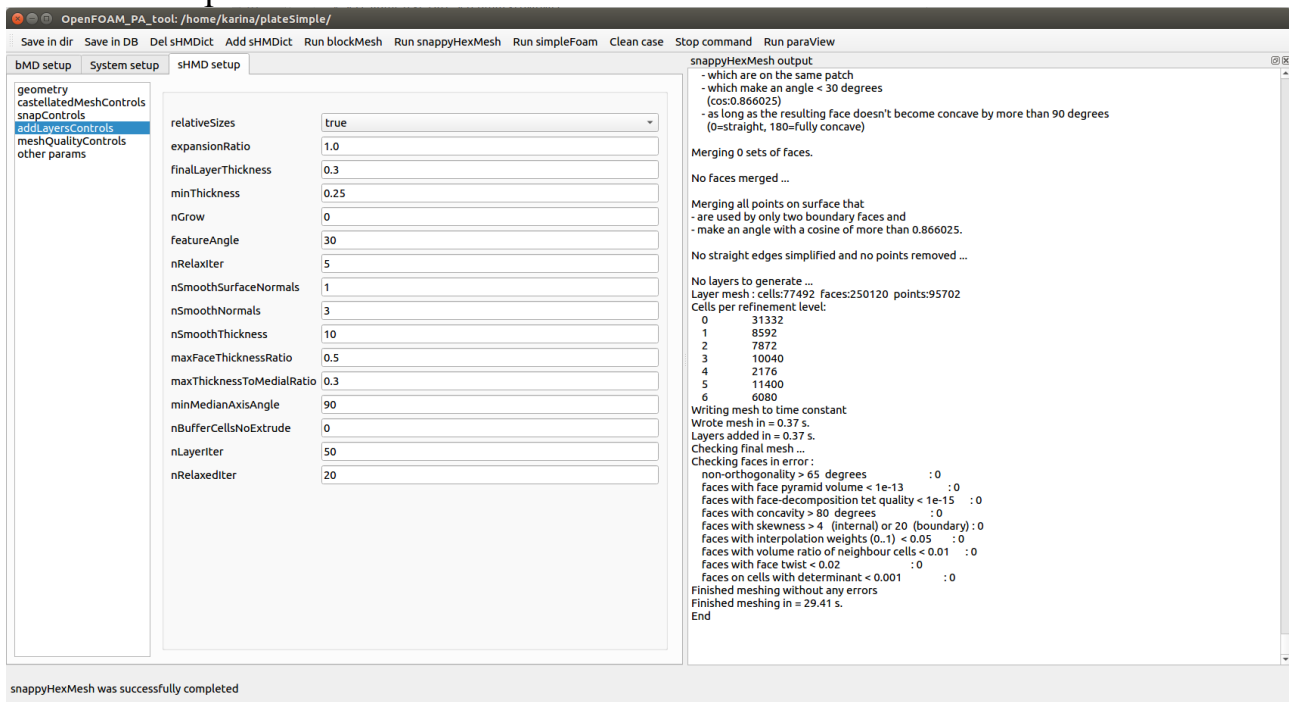


Рис. 10. Результат работы команды snappyHexMesh

Команда snappyHexMesh выполнялась без ошибок. Для проверки наличия построенной сетки запускается команда paraFoam, что показано на рис. 11.

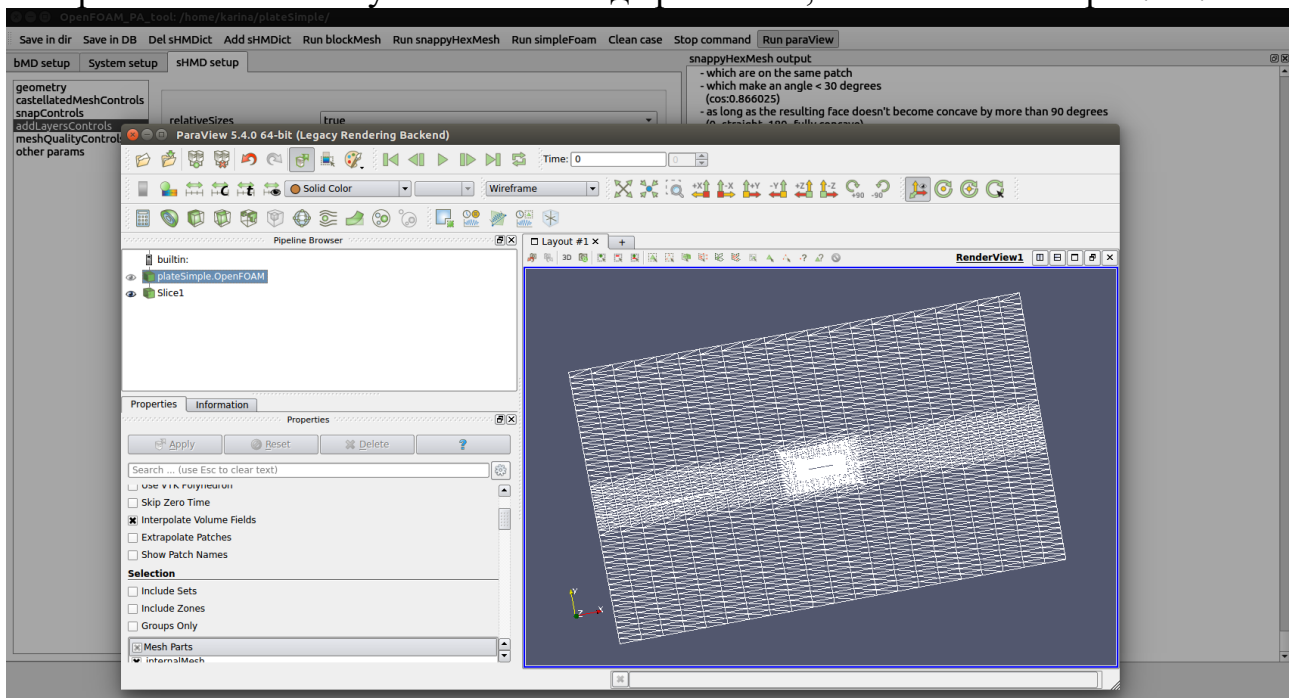


Рис. 11. Проверка построенной сетки с помощью ParaView

Проверка наличия сетки пройдена успешно, теперь подготовлен весь набор исходных данных для проведения расчёта обтекания тела потоком вязкой несжимаемой жидкости.

Таким образом, в рамках данной работы реализована базовая версия графической оболочки для подготовки исходных данных численного эксперимента на базе платформы OpenFOAM с помощью языка программирования Python 3.5 и фреймворка PyQt5 [23]. Программа представляет собой десктопное приложение с графическим интерфейсом, включающим в себя стартовое окно, основное окно с панелью управления, блоком подготовки исходных данных для расчетного случая, блоком отображения результатов и панелью вывода служебной информации. Данное приложение предоставляет возможность:

1) подготовки нового расчетного случая и открытия существующего для внесения изменений или проверки параметров. В рамках данной опции возможна подготовка расчетных случаев с несложной геометрией (когда внутри домена расположен простой объект, например, шар, пластина или конус) и указанием начальных условий для скорости и давления, а также базовых параметров решателя simpleFoam;

2) генерации расчётных сеток средствами OpenFOAM (с помощью встроенных утилит blockMesh и snappyHexMesh);

3) запуска и остановки процесса решения, запущенного посредством решателя simpleFoam. В процессе решения осуществляется подсчет давления и скорости при ламинарном режиме течения;

4) визуализации результатов моделирования с помощью инструмента постобработки ParaView, интегрированного в OpenFOAM.

Программный запуск консольных команд OpenFOAM реализуется посредством вызова bash-скриптов с помощью модуля subprocess языка Python

### Литература

1. OpenFOAM. URL: <https://openfoam.org> (дата обращения: 10.06.2022).
2. Панкратов И.А. Математическое моделирование реальных процессов в пакете OpenFOAM. Саратов.: Саратовский источник, 2019. 61 с.
3. HELYX-OS. URL: <https://engys.com/products/helyx-os> (дата обращения: 10.06.2022).
4. Salome. URL: <https://www.salome-platform.org> (дата обращения: 10.06.2022).
5. Visual-Environment for OpenFOAM. URL: <https://www.esi-group.com/products/computational-fluid-dynamics/visual-environment-for-openfoam> (дата обращения: 10.06.2022).
6. 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.
7. Читалов Д.И. Разработка графической оболочки для параллельных расчетов на базе платформы OpenFOAM // Программные продукты и системы. 2019. Т. 32. № 3. С. 433-443.

8. Иванов Д.А., Кожанов В.С. Автоматизация постпроцессинга при решении задач механики в CAD/CAE системах // Математика. Механика: сб. науч. тр. Саратов: Изд-во Сарат. ун-та, 2021. Вып. 23. С. 84-87.
9. The StL Format. URL: [http://www.fabbers.com/tech/STL\\_Format](http://www.fabbers.com/tech/STL_Format) (дата обращения: 12.06.2022).
10. Фаулер М. UML. Основы. М.: Символ-плюс, 2011. 192 с.
11. Киммел П. UML. Основы визуального анализа и проектирования. М.: ИТ Пресс, 2008. 272 с.
12. Advanced Bash-Scripting Guide. URL: [https://www.opennet.ru/docs/RUS/bash\\_scripting\\_guide/](https://www.opennet.ru/docs/RUS/bash_scripting_guide/) (дата обращения: 15.06.2022).
13. Bash-скрипты: начало. URL: <https://habr.com/en/company/ruvds/blog/325522/> (дата обращения: 15.06.2022).
14. Уилсон Д.Р., Редмонд Э. Семь баз данных за семь недель. Введение в современные базы данных и идеологию NoSQL. М.: ДМК Пресс, 2013. 384 с.
15. Блинков Ю.А. Панкратов И.А. Документоориентированное хранение и обработка библиографической информации // Актуальные направления научных исследований XXI века: теория и практика, 2017. Т. 5. № 8-1 (34-1). С. 63-66.
16. Блинков Ю.А. Панкратов И.А. Документо-ориентированное хранение и обработка научных публикаций // Математическое моделирование, компьютерный и натурный эксперимент в естественных науках, 2018. № 4. URL: [mathmod.esrae.ru/20-81](http://mathmod.esrae.ru/20-81) (дата обращения: 14.06.2022).
17. DB-Engines Ranking. URL: <https://db-engines.com/en/ranking> (дата обращения 10.06.2022).
18. Бэнкер К. MongoDB в действии. М.: ДМК-Пресс, 2012. 394 с.
19. Смирнов М.В., Толмасов Р.С. Графическая нотация моделирования документных баз данных // Открытое образование, 2021. Т. 25. № 5. С. 50-60.
20. Зимовец А.И., Хомоненко А.Д. Обоснование выбора модели хранения данных для системы мониторинга космического пространства // Автоматика на транспорте, 2019. Т. 5. № 2. С. 221-232.
21. Блинков Ю.А., Панкратов И.А. Хранение и обработка библиографической информации с помощью NoSQL // Актуальные направления научных исследований XXI века: теория и практика, 2017. Т. 5. № 7-1 (33-1). С. 247-249.
22. Курдаков Е.В., Панкратов И.А. Автоматизация построения отчётов о работе кафедры // Математическое моделирование, компьютерный и натурный эксперимент в естественных науках, 2021. № 1. URL: [mathmod.esrae.ru/33-124](http://mathmod.esrae.ru/33-124) (дата обращения: 14.06.2022).
23. Прохоренок Н.А., Дронов В.А. Python 3 и PyQt 5. Разработка приложений. СПб.: БХВ-Петербург, 2016. 832 с.