

Министерство науки и высшего образования Российской Федерации  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЯДЕРНЫЙ УНИВЕРСИТЕТ  
«МИФИ»

---

ИНСТИТУТ ЛАЗЕРНЫХ И ПЛАЗМЕННЫХ ТЕХНОЛОГИЙ  
КАФЕДРА №31 ПРИКЛАДНАЯ МАТЕМАТИКА

Отчет  
по научно-исследовательской работе на тему:

«Оптимизация канального радиатора»

*Выполнил: Есис А. И.*

*Руководитель проекта: Чмыхов М. А.*

г. Москва 2023

## Аннотация

Данная работа посвящена исследованию и оптимизации формы радиатора с применением трех программных инструментов: OpenFOAM, ParaView и SALOME Меса. Целью исследования является повышение эффективности радиатора путем оптимизации его геометрии.

В начале исследования используется SALOME Меса для построения геометрии и сетки радиатора. На втором этапе проводятся численные симуляции с использованием OpenFOAM. В результате симуляций получаются данные о тепловом и газодинамическом поведении радиатора. Наконец, результаты численных симуляций визуализируются с помощью ParaView. И делается вывод о наиболее оптимальной форме радиатора.

# Содержание

<b>1</b>	<b>Введение</b>	<b>4</b>
<b>2</b>	<b>Основная часть</b>	<b>5</b>
<b>3</b>	<b>Проблема с Генерацией Сетки в SALOME</b>	<b>15</b>
3.1	SubShapeAllIDs . . . . .	16
3.2	GetShapesOnBoxIDs . . . . .	16
3.3	GetShapesOnPlaneWithLocationIDs . . . . .	16
<b>4</b>	<b>Автоматизированный процесс генерации сетки и расчета</b>	<b>17</b>
<b>5</b>	<b>Особенности Фиксации и Движения Цилиндров</b>	<b>17</b>
<b>6</b>	<b>Первые Результаты и Интерполяция Поверхности</b>	<b>18</b>
<b>7</b>	<b>Заключение</b>	<b>20</b>
<b>8</b>	<b>Приложение А</b>	<b>22</b>
<b>9</b>	<b>Приложение Б</b>	<b>23</b>

# 1 Введение

OpenFOAM, ParaView и SALOME Меса являются мощными программными инструментами, широко применяемыми в области вычислительной гидрогазодинамики (CFD) и численного моделирования. Вместе они предоставляют комплексное решение для анализа и визуализации сложных физических процессов, таких как течение жидкостей, теплообмен, движение твердых тел и другие.

OpenFOAM (Open Field Operation and Manipulation) является свободным и открытым программным обеспечением для решения уравнений Навье-Стокса и других математических моделей, связанных с течением жидкостей и газов. OpenFOAM предоставляет широкий спектр методов решения, таких как конечно-разностные, конечно-объемные и конечно-элементные, что позволяет исследовать различные типы потоков и применять разные физические модели. Будучи свободно распространяемым и расширяемым, OpenFOAM предоставляет возможность настраивать и адаптировать код под конкретные задачи и требования.

ParaView является инструментом, который позволяет визуализировать результаты численных симуляций. Он предоставляет широкий спектр функций для создания визуализаций, включая 2D и 3D графику, построение контуров, срезов и анимаций. ParaView также поддерживает интерактивное взаимодействие с моделью, что позволяет анализировать данные в реальном времени, изменять параметры симуляции и осуществлять глубокий анализ результатов.

SALOME Меса является интегрированной средой для предварительной обработки геометрии и настройки расчетной сетки для OpenFOAM. Он предоставляет интуитивный пользовательский интерфейс, который облегчает создание и манипулирование сложными геометрическими моделями, а также настройку сеток с различной структурой. SALOME Меса также предлагает набор инструментов для проверки качества сетки и подготовки ее к последующим симуляциям в OpenFOAM [1].

Вместе OpenFOAM, ParaView и SALOME Меса образуют мощный комплект инструментов для моделирования и визуализации в области вычислительной гидрогазодинамики. Они предоставляют возможность проводить сложные численные симуляции, анализировать результаты и визуализировать данные, что помогает в понимании физических процессов и принятии информированных решений в различных областях, таких как авиация, автомобильная промышленность, энергетика и многое другое.

## 2 Основная часть

В данном проекте исследуется задача охлаждения нагретого тела с помощью установки радиатора.

В начале проекта геометрия радиатора строится с использованием графического интерфейса SALOME. Используя интуитивно понятный пользовательский интерфейс, создается геометрия радиатора. Этот этап позволяет получить исходную геометрию, которая будет использоваться для последующих анализов.

Затем, для повышения гибкости и автоматизации процесса, создается скрипт на языке Python, который генерирует геометрию радиатора. В этом скрипте можно устанавливать параметры радиатора, такие как расположение элементов. Это позволяет быстро создавать и изменять различные варианты геометрии радиатора для дальнейшего анализа и оптимизации.

Для исследования были использованы три различных варианта геометрии радиатора. Примеры геометрии:

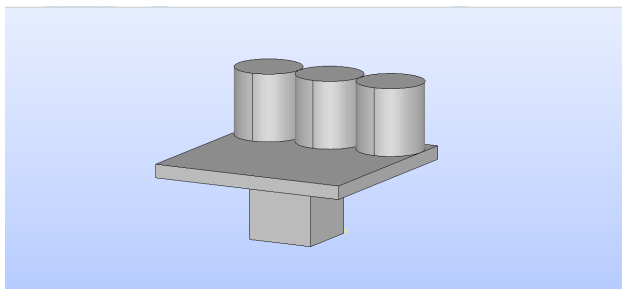


Рис. 1: Модель геометрии 1

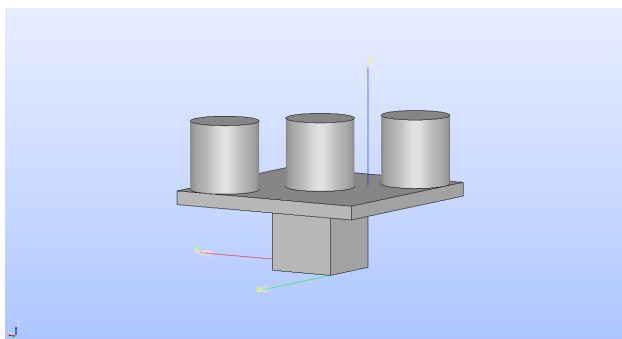


Рис. 2: Модель геометрии 2

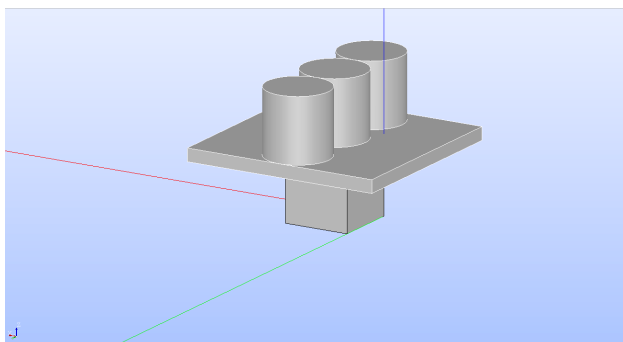


Рис. 3: Модель геометрии 3

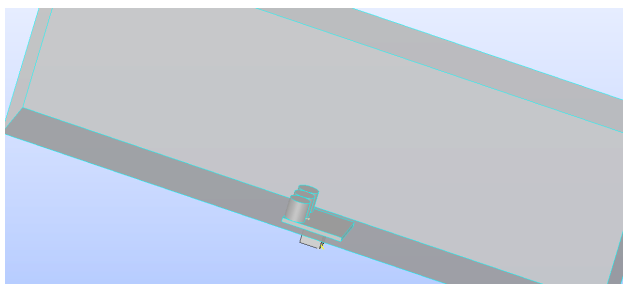


Рис. 4: Модель геометрии 1

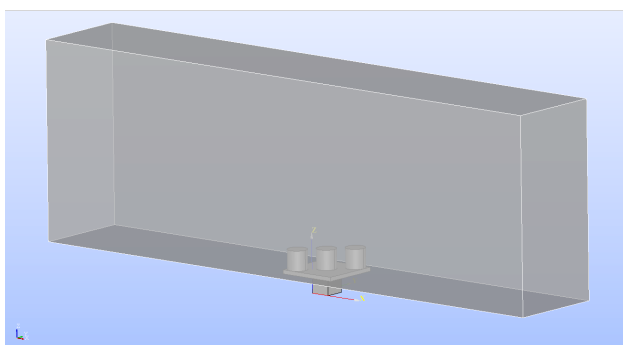


Рис. 5: Модель геометрии 2

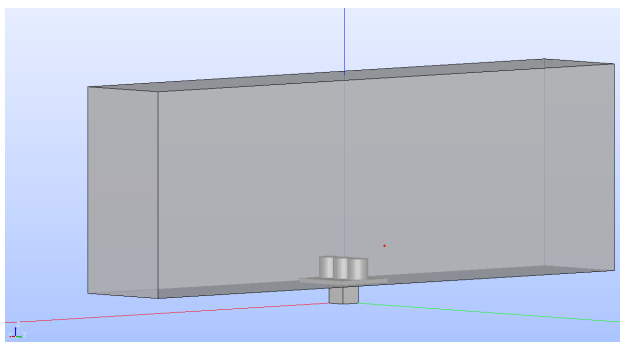


Рис. 6: Модель геометрии 3

В данных моделях части радиатора (3 цилиндрических элемента) являются подвижными и могут быть сдвинуты по оси Ох. Их радиус основания равен 5 мм, а высота 10 мм. Размеры подложки радиатора по ширине и длине 30 мм, а по высоте 2 мм. Нагреватель представляет собой куб с длиной ребра 10 мм и объемной плотностью источников тепла  $2.94 \cdot 10^7$  Вт/³. Скорость потока воздуха 5.6 м/с. Воздух находится в параллелепипеде размерами 300 мм по длине, 50 мм по ширине и 100 мм по высоте. Начальная температура –296.9 К. Параметры материалов приведены в таблице 1 [2].

	Нагреватель	Радиатор	Воздух
Плотность [кг/³]	1280	2700	1.196
Ср [Дж/кг*К]	1004	900	1005
Коэффициент теплопроводности [Вт/м*К]	80	200	
Молекулярная масса [г/моль]	50	27	28.9
Вязкость [кг/м*с]			$1.8 \cdot 10^{-5}$
Число Прандтля			0.7

Таблица 1: Параметры задачи

Затем по модели была построена сетка с уплотнением в области радиатора и нагревателя и произведено разбиение на регионы:

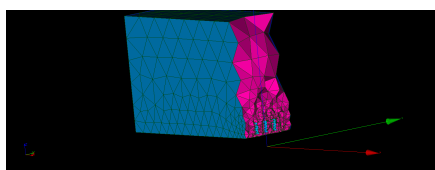


Рис. 7: Сетка модели 1

Для численного решения задачи используется решатель chtMultiRegionFoam. Он применяется для расчета теплообмена между жидкостью/газом и твердым телом. А также для моделирования сложных задач, связанных с теплопередачей и теплообменом в многорегиональных системах [3]. Для каждого региона задавались начальные и граничные условия. Были также заданы дополнительные функции для анализа [2]. Для моделей были получены следующие результаты распределений по температурам:

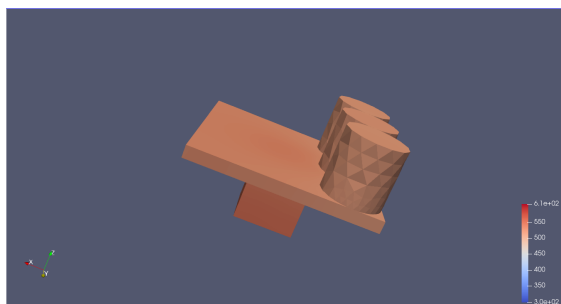


Рис. 8: Распределение температуры для модели 1

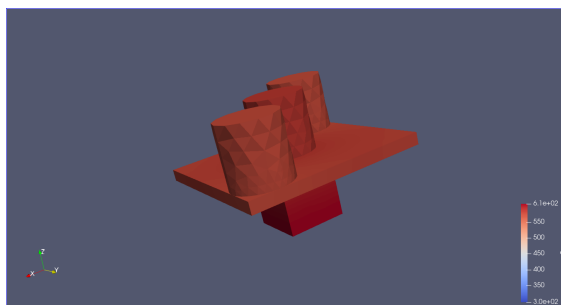


Рис. 9: Распределение температуры для модели 2

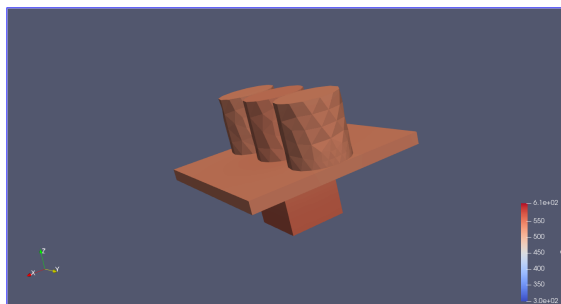


Рис. 10: Распределение температуры для модели 3



Были получены следующие результаты для моделей:

**1. Первая модель:**

- Средняя температура радиатора: 508.381 K
- Средняя температура нагревателя: 533.363 K
- Средняя температура интерфейса между нагревателем и радиатором: 521.537 K

**2. Вторая модель:**

- Средняя температура радиатора: 555.23 K
- Средняя температура нагревателя: 576.306 K
- Средняя температура интерфейса между нагревателем и радиатором: 564.451 K

**3. Третья модель:**

- Средняя температура радиатора: 519.325 K
- Средняя температура нагревателя: 537.741 K
- Средняя температура интерфейса между нагревателем и радиатором: 525.862 K

Построены графики температуры в нагревателе:

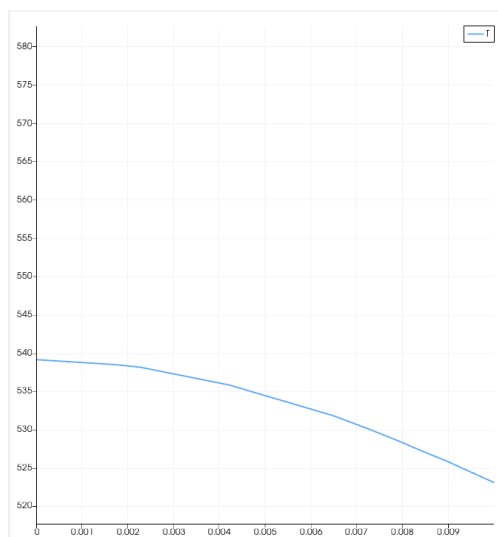


Рис. 11: Распределение температуры в нагревателе для модели 1

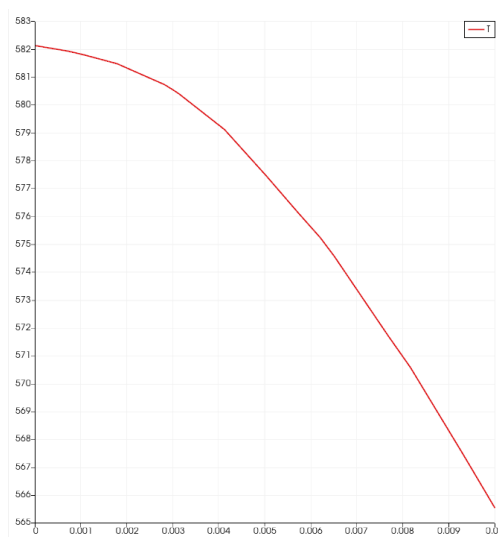


Рис. 12: Распределение температуры в нагревателе для модели 2



Рис. 13: Распределение температуры в нагревателе для модели 3

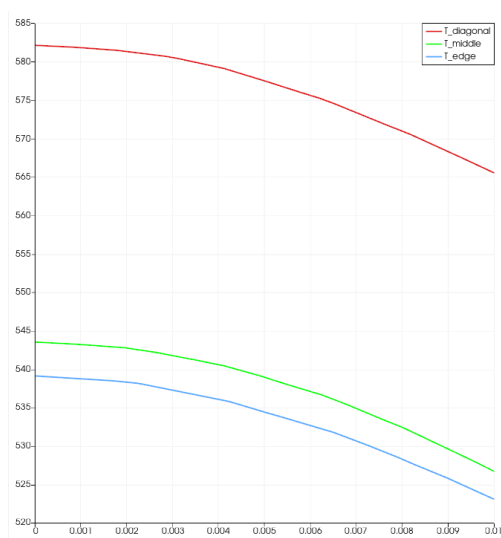


Рис. 14: Распределение температур в нагревателях для всех моделей

Затем было проведено 2000 итераций расчета. Для отслеживания сходимости во время проведения расчета были построены следующие графики (пример для модели 1):

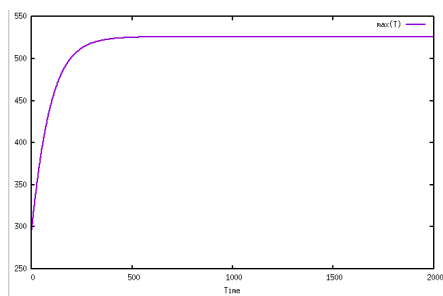


Рис. 15: Зависимость максимальной температуры радиатора от шага

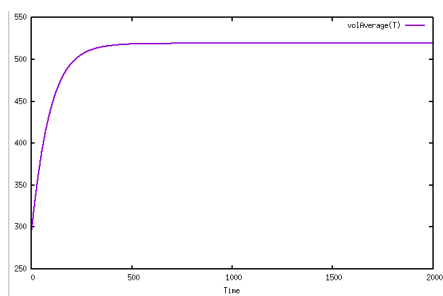


Рис. 16: Зависимость средней температуры радиатора от шага

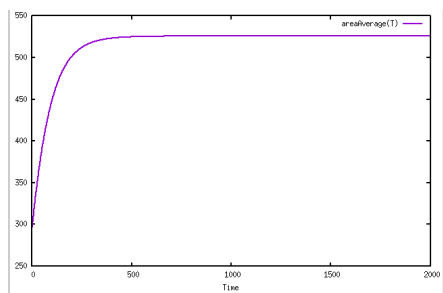


Рис. 17: Зависимость средней температуры интерфейса между нагревателем и радиатором от шага

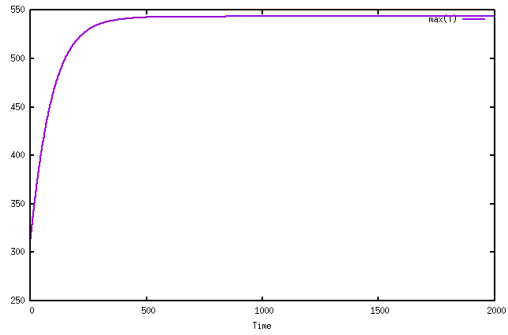


Рис. 18: Зависимость максимальной температуры нагревателя от шага

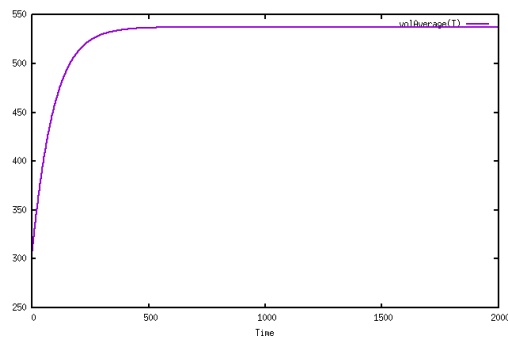


Рис. 19: Зависимость средней температуры нагревателя от шага

Далее были построены графики распределений температуры в центральном сечении:

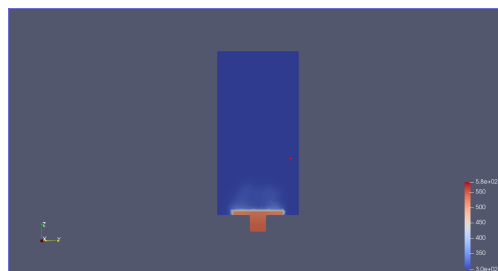


Рис. 20: Распределение температуры для модели 1

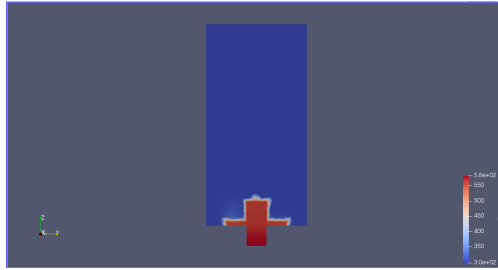


Рис. 21: Распределение температуры для модели 2

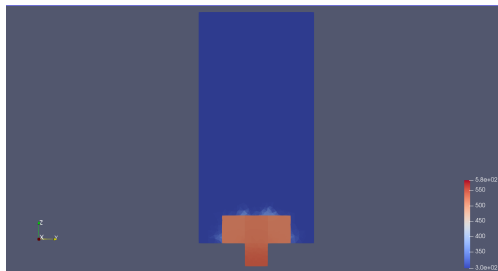


Рис. 22: Распределение температуры для модели 3

И графики распределений скоростей:

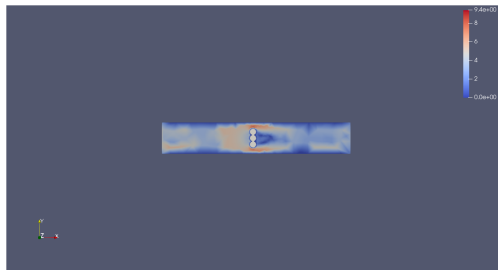


Рис. 23: Распределение скоростей для модели 1

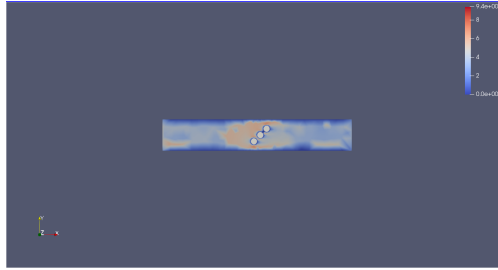


Рис. 24: Распределение скоростей для модели 2

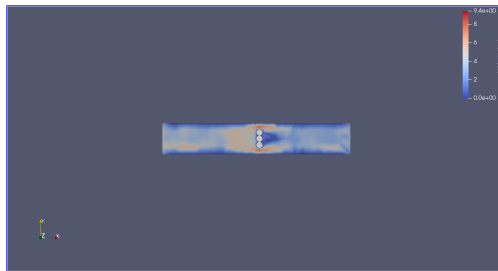


Рис. 25: Распределение скоростей для модели 3

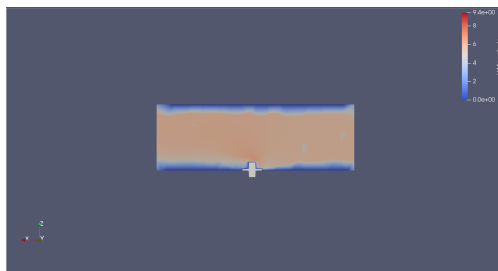


Рис. 26: Распределение скоростей для модели 1

### 3 Проблема с Генерацией Сетки в SALOME

В ходе нашего исследования мы столкнулись с проблемой при регенерации геометрии в SALOME, используя скрипты Python API. Проблема заключалась в том, что при каждой попытке обновления геометрии с новыми данными, мы получали сетку с ошибками из-за нарушения идентификаторов (ID), так как они изменялись.

Для решения этой проблемы мы обратились к Python API в SALOME и использовали следующие методы:

### 3.1 SubShapeAllIDs

Функция `SubShapeAllIDs` в Python API SALOME используется для получения всех идентификаторов (ID) подформ в геометрии. Это полезно, когда необходимо провести операции с каждой подформой в модели.

Пример использования:

```
subshape_ids = geompy.SubShapeAllIDs(main_shape)
print(f"ID подформ: {subshape_ids}")
```

В данном примере `main_shape` - это главная форма в вашей геометрии.

### 3.2 GetShapesOnBoxIDs

Функция `GetShapesOnBoxIDs` возвращает идентификаторы форм, которые содержатся внутри заданного объема, определенного прямоугольным параллелепипедом.

Пример использования:

```
box = geompy.MakeBox(0, 0, 0, 10, 10, 10)
shapes_inside_box_ids = geompy.GetShapesOnBoxIDs(main_shape, box)
print(f"ID форм внутри прямоугольного параллелепипеда: {shapes_inside_box_ids}")
```

Здесь `main_shape` - это опять же главная форма, а `box` - созданный прямоугольный параллелепипед.

### 3.3 GetShapesOnPlaneWithLocationIDs

Функция `GetShapesOnPlaneWithLocationIDs` возвращает идентификаторы форм, которые пересекают заданную плоскость.

Пример использования:

```
plane = geompy.MakePlane(0, 0, 1, 0)
shapes_on_plane_ids = geompy.GetShapesOnPlaneWithLocationIDs(main_shape, plane)
print(f"ID форм, пересекаемых плоскостью: {shapes_on_plane_ids}")
```

В этом примере `main_shape` - ваша главная форма, а `plane` - созданная плоскость.

Используя эти методы, мы смогли стабилизировать процесс генерации сетки, обеспечивая постоянство идентификаторов форм даже при обновлении геометрии. Это решение позволило нам успешно использовать скрипты Python API для регенерации геометрии в дальнейших этапах нашего исследования.



## 4 Автоматизированный процесс генерации сетки и расчета

В данном разделе рассматривается процесс автоматизации генерации сетки в программе SALOME и выполнения расчетов в OpenFOAM для различных комбинаций параметров. Ключевыми параметрами, подлежащими варьированию, являются `shift_first_cylinder` и `shift_second_cylinder`.

Переменная `shift_first_cylinder` представляет собой сдвиг первого цилиндра в радиаторе, а `shift_second_cylinder` — сдвиг второго цилиндра. Эти параметры влияют на геометрию радиатора и, следовательно, на условия теплообмена в системе.

Скрипт создает уникальное имя для каждой комбинации параметров и затем копирует исходный кейс в новую директорию с уникальным именем. После этого SALOME запускается в режиме командной строки для выполнения генерации сетки с учетом новой геометрии, заданной параметрами сдвига цилиндров. Далее запускаются соответствующие bash-скрипты для выполнения расчета сетки и кейса в OpenFOAM.

После завершения расчета, автоматически извлекаются результаты. Сценарий ищет и анализирует файл с данными теплообмена в постпроцессинговой директории. Это позволяет собирать и систематизировать конечные результаты для каждой конфигурации геометрии. Такой подход обеспечивает эффективный анализ влияния различных параметров на теплоотдачу системы.

## 5 Особенности Фиксации и Движения Цилиндров

Необходимо отметить, что в проведенных исследованиях у нас было три цилиндра в системе. Тем не менее, третий цилиндр был зафиксирован в положении  $(0, 0)$ , и двигались только два оставшихся цилиндра. Такой выбор сделан с целью упростить визуализацию и анализ результатов.

Такой подход облегчает анализ и интерпретацию результатов, обеспечивая более ясное представление о влиянии конкретных параметров на эффективность теплообмена в рассматриваемой системе.

## 6 Первые Результаты и Интерполяция Поверхности

На основе автоматизированного процесса были получены первые результаты, охватывающие 25 точек варьирования параметров. Исследуемые параметры `shift_first_cylinder` и `shift_second_cylinder` изменялись от 0 до 20 с шагом 5. Эти значения представляют различные комбинации сдвигов цилиндров в радиаторе, что отражает влияние геометрических параметров на теплоотдачу системы.

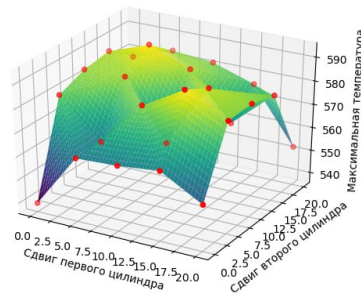


Рис. 27: Зависимость максимальной температуры нагревателя от сдвигов цилиндров

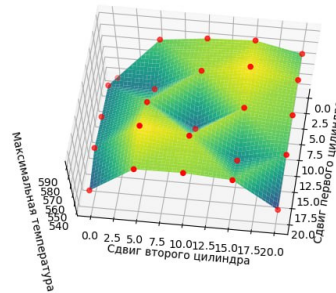


Рис. 28: Зависимость максимальной температуры нагревателя от сдвигов цилиндров

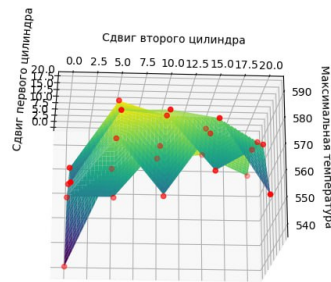


Рис. 29: Зависимость максимальной температуры нагревателя от сдвигов цилиндров

Для улучшения визуализации и наглядности полученных данных была проведена интерполяция поверхности. Интерполированная поверхность дает представление о поведении системы в пространстве параметров и позволяет выявить области оптимальных значений для исследуемых параметров.

Этот этап анализа предоставляет первичное представление о влиянии сдвига цилиндров на теплообмен в системе.

## 7 Заключение

Результаты анализа указывают на следующие выводы: вторая модель имеет наивысшую среднюю температуру радиатора, нагревателя и интерфейса между ними, что может указывать на менее эффективное охлаждение. Первая и третья модели демонстрируют более низкие значения средних температур радиатора, нагревателя и интерфейса. Таким образом, основываясь на полученных результатах, можно сделать вывод о наибольшей эффективности охлаждения и лучшей теплоотдаче в первой модели, где наблюдается наименьшая средняя температура нагревателя, указывающая на более эффективную передачу тепла. Результаты исследования подтверждают, что форма радиатора влияет на его эффективность и теплоотдачу. Модель 1 с определенным распределением цилиндров обеспечивает наилучшие результаты в задаче охлаждения нагретого тела. Данный результат связан с тем, что форма и расположение цилиндров в модели 1 позволяют более равномерно распределить тепловую нагрузку и обеспечить эффективное охлаждение, поскольку цилиндры имеют прямой контакт между собой. Во второй модели, где отсутствует касание между цилиндрами, наблюдается наихудший результат среди рассмотренных моделей.

## Список литературы

- [1] Introduction to Salome Platform — GUI 9.5.0 documentation. URL: <https://docs.salome-platform.org/latest/gui/GUI/introduction.html>.
- [2] Conjugate heat transfer CFD tutorial for plate fin heat sink. 2020. URL: <http://rgdoi.net/10.13140/RG.2.2.26603.36640>.
- [3] OpenFOAM: User Guide: chtMultiRegionFoam. URL: <https://www.openfoam.com/documentation/guides/latest/doc/guide-applications-solvers-heat-transfer-chtMultiRegionFoam.html>.

## 8 Приложение А

Проект можно найти на github ([github.com/AlexEsn/FOAM\\_project](https://github.com/AlexEsn/FOAM_project)), где содержатся все скрипты и кейсы.

## 9 Приложение Б

В процессе работы был найден баг в SALOME-9.9.0: при дампе Python скрипта с сеткой создается файл с ошибкой.

```
try:
    Mesh_1.ExportUNV(r'C:/File', 0)
except:
    print('ExportUNV() failed. Invalid file name?')
```

Рис. 30: Ошибка экспорта сетки

Для её решения достаточно удалить лишние символы:

```
try:
    Mesh_1.ExportUNV(r'C:/File')
except:
    print('ExportUNV() failed. Invalid file name?')
```

Рис. 31: Решение ошибки экспорта сетки