

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

ИНСТИТУТ ЛАЗЕРНЫХ И ПЛАЗМЕННЫХ ТЕХНОЛОГИЙ
КАФЕДРА №31 ПРИКЛАДНАЯ МАТЕМАТИКА

Отчет

по проектной практике на тему:

«Изучение навыков работы в программе SALOME. Изучение работы с командами консоли. Написание скриптов на языке Python.»

Авторы: Есис А. И., Павленок Л. Л.

Руководитель проекта: Чмыхов М. А.

Содержание

1	Проделанная работа	5
2	Дальнейшая работа	14

Аннотация

3D-моделирование стало неотъемлемой частью нашей жизни. Сегодня оно широко используется в сфере маркетинга, архитектурного дизайна и кинематографии, не говоря уже о промышленности. 3D-моделирование позволяет создать прототип будущего сооружения, коммерческого продукта в объемном формате.

Сегодня невозможно представить 2 крупные отрасли без применения трехмерных моделей. Это — промышленность и индустрия развлечений. С индустрией развлечений мы сталкиваемся почти каждый день. Это фильмы, анимация и компьютерные игры. С помощью одного и того же принципа — полигонального моделирования созданы многие виртуальные миры и персонажи. Полигональное моделирование происходит путем манипуляций с полигонами в пространстве. Вытягивание, перемещение, вращение, и.т.д. Однако невозможно контролировать необходимые зазоры, сечения, учесть физические свойства материала и технологию изготовления, используя полигональное моделирование. Для таких изделий используются методы промышленного проектирования.

Цель проекта: научиться строить 3D модели с помощью визуального интерфейса и Python скриптов, создавать расчетную сетку моделей и вычислять поведения объектов в разных ситуациях.

Задачи:

1. Научиться работать с визуальным интерфейсом программы SALOME
2. Построить 3D модели с помощью визуального интерфейса программы SALOME
3. Построить 3D модели с помощью Python скриптов
4. Наложить сетку на изучаемую модель
5. Построить математическую модель исследуемого объекта и рассчитать его поведение в различных средах

1 Прделанная работа

На данном этапе выполнения проета была изучена литература, на основании которой был составлен следующий обзор: САПР (Система Автоматизированного Проектирования). Современные CAD-подсистемы, входящие в состав интегрированных CAD/CAM/CAE-систем — поддержанное компьютером конструирование/изготовление/инженерная деятельность), и системы твердотельного параметрического моделирования механических объектов, отражающие последние достижения инженерной компьютерной графики, представляют собой наиболее важные разработки в области новых технологий по автоматизации деятельности инженеров, конструкторов и технологов. По уровню возможностей САПР условно разделяются на три уровня:

1. САПР нижнего уровня — это, чаще всего, программы для двумерного проектирования.
2. САПР среднего уровня — позволяют дополнительно создавать трехмерные параметрические модели и выполнять проверочные расчеты деталей и сборок.
3. САПР верхнего уровня — обеспечивают потребности практически всех областей проектирования, от разработки изделий и оснастки до проведения сложных инженерных расчетов и изготовления изделий[1].

В данной работе мы используем САПР среднего уровня, которой является Salome-MECA. Изначально задуманная в качестве связующего ПО CAD-CAE, она включает в себя разнообразные модули, используемые в приложениях численного моделирования — от моделирования в САПР до параллельных вычислений. Одним из важнейших преимуществ Salome-MECA является то, что её исходный код открыт и бесплатно распространяется на условиях GNU Lesser General Public License. Это означает, что Salome может расширяться за счет сторонних свободных или коммерческих модулей. Внутренним языком платформы является Python, причем в самой платформе имеется встроенная консоль, которая может быть использованна для выполнения пользовательских сценариев и автоматизации обработки множества типовых задач (пакетной обработки)[2]. Платформа Salome, как и любая CAE-система, состоит из четырех модулей: решателя (Solver), построителя сетки (Mesh), модуля построения геометрии (Geometry), модуля обработки и представления результатов (Post-processing).

Метод конечных элементов лежит в основе некоторых САЕ-пакетов . Идея этого метода заключается в замене непрерывной функции, описывающей изучаемое явление или процесс, дискретной моделью, которая строится на базе множества кусочно-непрерывных функций, определенных на конечном числе подобластей. Каждая такая подобласть конечна и представляет собой часть (элемент) всей области, поэтому их называют конечными элементами. Исследуемая геометрическая область разбивается на элементы так, чтобы на каждом из них неизвестная функция аппроксимировалась пробной функцией. Такое разбиение называется расчетной сеткой. Создание хорошей расчетной сетки также представляет собой нетривиальную задачу. Это связано с тем, что реальные детали машин имеют сложную геометрию и необходимо разделить их на такие элементы, чтобы приближенные решения не сильно отличались от точных. Поэтому, кроме самих САЕ-пакетов, существует большое число приложений, выполняющих всего одну важную функцию: построение расчетной сетки [3]. В данной работе наибольший интерес представляют модуль Geometry и Mesh, поскольку для них предусмотрено написание скриптов.

Любая задача моделирования начинается с создания геометрической модели изучаемого объекта – определения его формы. Другими словами, необходимо задать границы той части пространства, поведение которой необходимо исследовать. Геометрическая модель может представлять собой двумерный или трехмерный чертеж.

Геометрический модуль SALOME разработан для:

- Построения и оптимизации геометрических моделей, используя широкий набор функций автоматизированного проектирования
- Создание элементарных геометрических объектов
- Построение примитивов
- Построение форм
- Генерирование сложных форм
- Работа с группами
- Геометрические булевы (логические) операции
- Геометрические преобразования
- Построение блоками

В геометрический модуль можно как экспортировать, так и импортировать геометрические объекты следующих форматов: BREP, IGES, STEP. Механизмы импорта и экспорта осуществлены через дополнения (плагины), что дает возможность расширить диапазон доступных форматов, добавляя другие приложения-дополнения (плагины) (такие как CATIA 5)[4].

Любой реальный исследуемый объект чаще всего обладает достаточно сложной формой. И описать его поведение при наложении внешних нагрузок, как цельное, достаточно сложно хотя бы потому, что любой объект чаще всего имеет в своем составе хотя бы несколько элементов. Следовательно, нужно записать точные математические выражения для граничных условий на поверхности каждого типа, а затем совместить их.

Можно использовать другой алгоритм, разбивая всю деталь на очень мелкие фрагменты определенной формы (конечные элементы): треугольники, четырехугольники, шестиугольники и т.д. для 2D-объектов и тетраэдры, гексаэдры и т.д. для 3D-объектов. Граничные условия для каждого такого фрагмента будут иметь достаточно простой вид. Более того, из-за малых размеров элемента, решение для него можно аппроксимировать при помощи элементарных функций, например, линейным или квадратичным полиномом. Используя данный алгоритм, мы получим систему алгебраических уравнений, для решения которой существует большое количество легко реализуемых методов.

При разбиении объекта на малые элементы сначала необходимо определить точки, которые будут вершинами элементов. Эти точки называют узлами (node) расчетной сетки или 0D-элементами. Далее, на основе выбранных алгоритмов и геометрии базовых элементов, узлы соединяются прямыми линиями, называемыми ребрами (edge) или 1D-элементами. Область, заключенная между несколькими ребрами и не содержащая ни одного узла или ребра, называется гранью (face) или 2D-элементом. Обычно грани образуются тремя или четырьмя ребрами. Область, заключенная между несколькими гранями и не содержащая ни одного узла, ребра или грани (части грани) называется объемным (volume object) или 3D-элементом.

Разбиение объекта на элементы в Salome выполняется по выбираемым пользователем алгоритмам и на основе параметров, оформляемых в виде гипотез [5].

Модуль MESH SALOME предназначен для[2]:

- Импорта и экспорта сеток в формате MED

- Сцепление геометрических моделей, ранее созданных или импортированных компонентом Geometry
- Модификации в местном масштабе сгенерированных сеток
- Суммирование узлов и элементов (Addition of nodes and elements)
- Удаления узлов и элементов (Removal of nodes and elements)
- Контроля качества сеток, базирующихся на ряде определенных критериев

Чего нам удалось добиться в ходе выполнения проекта: Были построены 3D модели различных деталей различной сложности, что позволило научиться пользоваться визуальным интерфейсом программы. Модели строились по возрастанию сложности. От простых геометрических объектов, таких как куб, цилиндр и т. д. до сложных фигур включающих в себя множество простых объектов. Сложные детали были смоделированы по изображению детали с заданными размерами.

Простые модели:

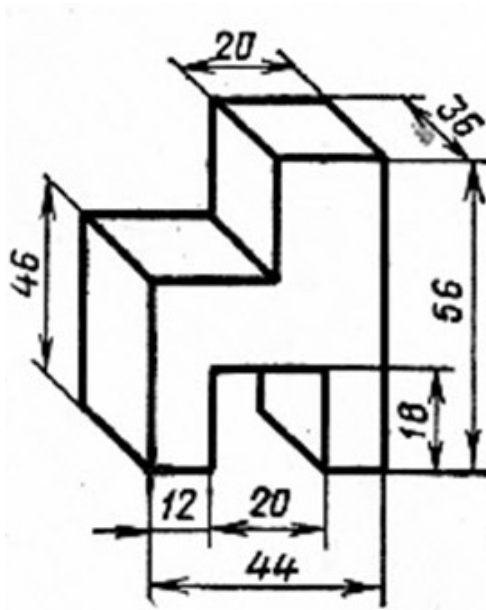


Рис. 1: Рисунок модели

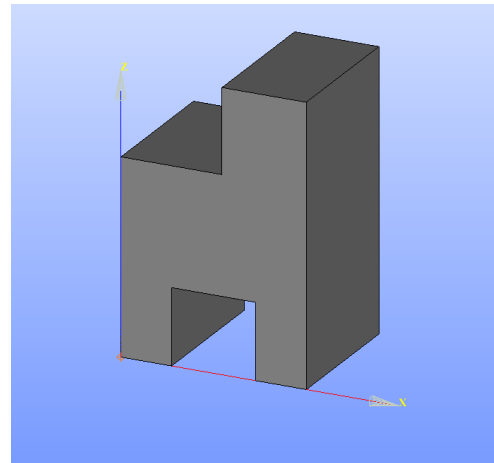


Рис. 2: Модель

Сложные модели:

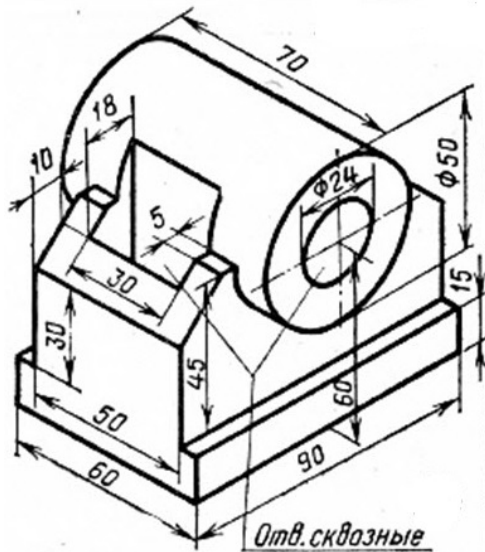


Рис. 3: Рисунок модели

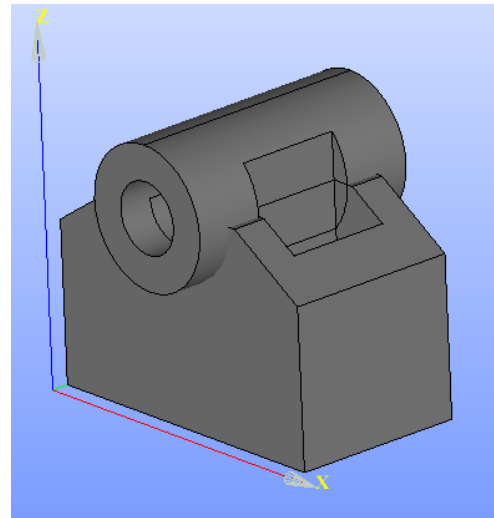


Рис. 4: Модель

Далее мы приступили к построению моделей с помощью Python скриптов. Это более сложная и интересная задача, с синтаксисом языка мы были ознакомлены, поэтому с этим проблем не возникло. Сначала мы исследовали уже построенные детали, смотрели как они выглядят в виде Python скрипта, после чего начали сами экспериментировать, что заняло много времени, но это было необходимо, так как дальнейшая работа основывалась на этом. Переходя от простых деталей к более сложным нам удалось достигнуть результата и мы научились строить модели деталей скриптами. Построение детали скриптами позволяет автоматизировать многие задачи и экономит время.

С помощью данного кода из скрипта

```
import sys
import salome
salome.salome_init()
import salome_notebook
notebook = salome_notebook.NoteBook()
import GEOM
from salome.geom import geomBuilder
import math
```



```

import SALOMEDS

geompy = geomBuilder.New()

#####

#####

##          Begin of NoteBook variables section          ##
#####

notebook.set("Number_Boxs", 50)
notebook.set("Length_Box1", 1000)
notebook.set("Widht_Box1", 1000)
notebook.set("Height_Box1", 100)

Number_Boxs = notebook.get("Number_Boxs")
Length_Box1 = notebook.get("Length_Box1")
Widht_Box1 = notebook.get("Widht_Box1")
Height_Box1 = notebook.get("Height_Box1")
#####

##          End of NoteBook variables section          ##
#####

Height_Boxs = Height_Box1 * 5
Length_Boxs = Length_Box1/Number_Boxs/2
Widht_Box1 = Widht_Box1

Boxs = [ ]

Translations = [ ]

O = geompy.MakeVertex(0, 0, 0)
OX = geompy.MakeVectorDXDYDZ(1, 0, 0)
OY = geompy.MakeVectorDXDYDZ(0, 1, 0)
OZ = geompy.MakeVectorDXDYDZ(0, 0, 1)
geompy.addToStudy( O, 'O' )
geompy.addToStudy( OX, 'OX' )
geompy.addToStudy( OY, 'OY' )
geompy.addToStudy( OZ, 'OZ' )
#####

Box_1 = geompy.MakeBoxDXDYDZ(Length_Box1, Widht_Box1, Height_Box1)
Box_1 = geompy.addToStudy( Box_1, 'Box_1' )

```

```

i = Number_Boxs
while i >= 0:
    Boxs.append(geompys.MakeBoxDXDYDZ(Length_Boxs, Widht_Box1, Height_Boxs))
    i = i - 1

```

```

i = Number_Boxs
j = 0
while i >= 0:
    Translations.append(geompys.MakeTranslation(Boxs[i], j, 0, 0))
    j = j + (Length_Boxs * 2)
    i = i - 1

```

```

i = Number_Boxs
while i >= 0:
    geompys.addToStudy( Translations[i], 'Translations' )
    i = i - 1

```

```

if salome.sg.hasDesktop():
    salome.sg.updateObjBrowser()

```

```

gg = salome.ImportComponentGUI("GEOM")
gg.createAndDisplayGO(Box_1)
gg.setDisplayMode(Box_1,1)
gg.setVectorsMode(Box_1, 1)
gg.setVerticesMode(Box_1, 1)
gg.setNameMode(Box_1, 1)

```

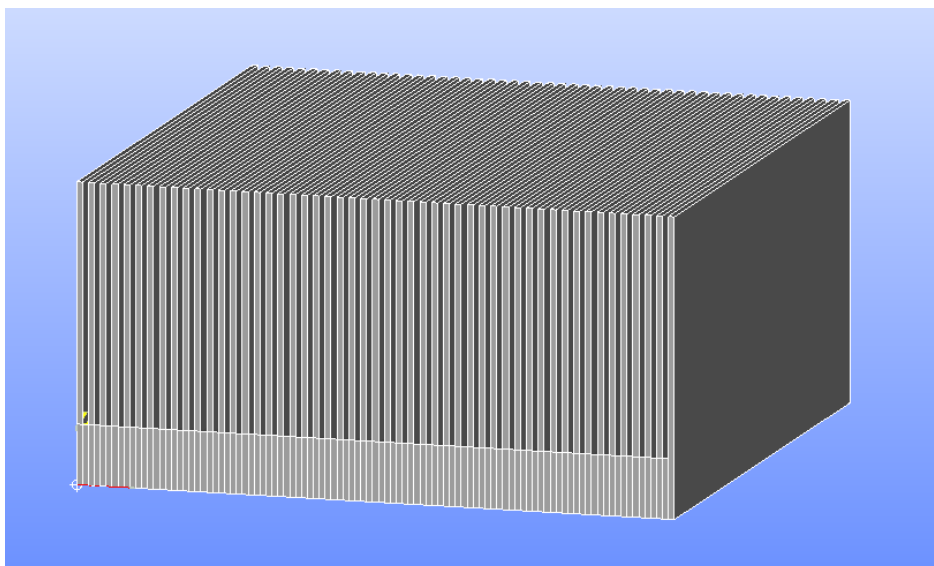


Рис. 5: Можно построить такую фигуру

После того, как мы научились строить 3D модели деталей, мы перешли к накладыванию сетки. Возникли сложности с подбором таких параметров, как: параметры гипотезы, параметры разбиения. Но поле нескольких экспериментов стало понятно, как подбирать данные параметры. И наконец мы перешли к построению самой математической модели. На данном этапе мы столкнулись с проблемой отсутствия открытого программного обеспечения для ОС Windows, решили эту проблему используя ОС Linux на виртуальной машине и открытое программное обеспечение OpenFOAM. С его помощью мы моделировали распространение тепла в твердом теле с учётом тепловых потерь. В качестве твердого тела была взята кастрюля, и смоделировано распространение тепла от горячего корпуса к ручкам. Результаты проделанной работы можно увидеть на картинках ниже:

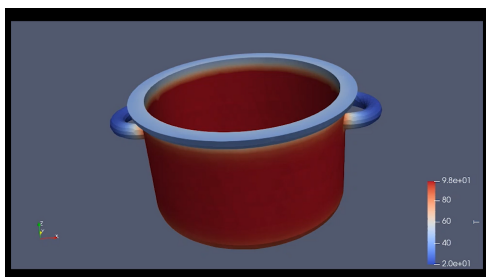


Рис. 6: 1-ая секунда

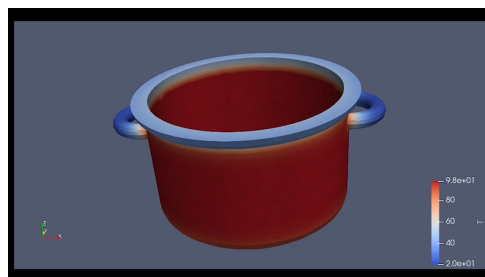


Рис. 7: 5-ая секунда

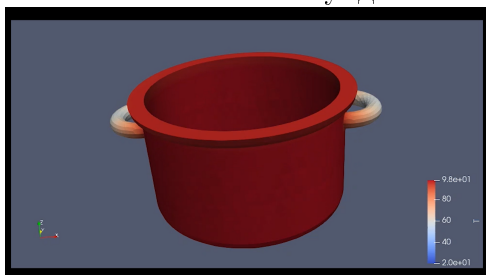


Рис. 8: 10-ая секунда



Рис. 9: 15-ая секунда

В ходе работы мы использовали систему контрол версий git, а также веб-сервис для хостинга GitHub, что было очень удобно в связи с дистанционным созданием проекта. Также применили программное обеспечение для управления библиографической информацией «Mendeley», в ней хранится вся найденная литература по нашей теме, которую в будущем мы сможем использовать для расширения проекта.

2 Дальнейшая работа

В дальнейшем планируется научиться накладывать сетку и строить математические модели без использования визуального интерфейса, а только при помощи Python скриптов. Наша конечная цель создать модель, которая будет строить оптимальную форму радиатора, находящегося в жидкой среде, учитывая характер движения жидкости, ее вязкость и количество. В качестве алгоритма подбора оптимальной формы планируется использовать генетические алгоритмы.

Список литературы

- [1] Большаков В., Бочков А. Основы 3D-моделирования. Изучаем работу в AutoCAD, КОМПАС-3D, SolidWorks, Inventor. СПб.: Питер, 2013.
- [2] Introduction to Salome Platform — GUI 9.5.0 documentation. URL: <https://docs.salome-platform.org/latest/gui/GUI/introduction.html>.
- [3] Моделирование в САЕ Salome: Часть 1. Общие принципы. URL: https://www.ibm.com/developerworks/ru/library/l-salome_01/index.html.
- [4] Документация Salome Geometry. URL: <http://laduga.ru/salome.html>.
- [5] Моделирование в САЕ Salome: Часть 3.1. Модуль Mesh. Построение двумерной сетки. URL: https://www.ibm.com/developerworks/ru/library/l-salome_03_01/index.html.