

Практическое занятие 2.1 Сочленения в опорно-двигательном аппарате человека

21 февраля 2022 г.

Содержание

1	Введение	1
2	Цель	1
3	Модель	1
4	Практическая часть - программирование в Jupyter Notebook	2
4.1	Подготовка окружения выполнения	2
4.1.1	Установка Anaconda	2
4.1.2	Установка зависимостей	3
4.1.3	Установка Gmsh	3
4.2	Пошаговая инструкция	4
4.2.1	Создание геометрии в Gmsh	4
4.2.2	Создание геометрии в текстовом режиме	9
4.2.3	Создание сетки в Gmsh	10
4.2.4	Скрипт Python для создания входных файлов	11
4.2.5	Визуализация модели	15
5	Заключение	17

1 Введение

Лабораторная работа по теме 2.1 "Сочленения и рычаги в опорно-двигательном аппарате человека" курса М1.Б.4 "Современные проблемы биомедицинской и экологической инженерии" для подготовки магистров по направлению 12.04.04 "Биотехнические системы и технологии".

На лекциях была рассмотрена тема анализа влияния дегенерации диска на механическое поведение поясничного двигательного сегмента с использованием метода конечных элементов. Была проиллюстрирована элементная сетка двигательного сегмента L3/L4 со здоровым и сильно дегенерированным дисками. Практическое занятие позволяет на практике освоить моделирование и исследование диска методом конечных элементов.

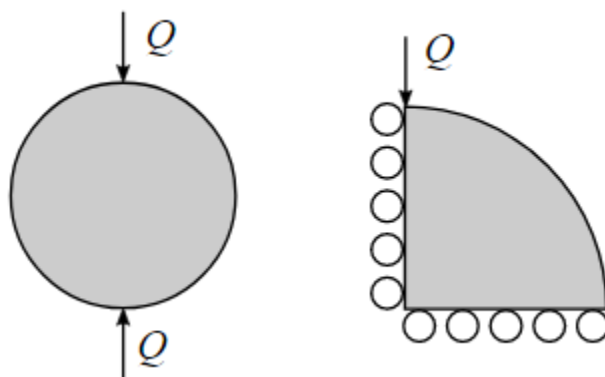
2 Цель

Получить навыки применения практических данных в построении графического материала для визуализации научных данных и результатов расчетов для углубления компетенции представлять современную научную картину мира, выявлять естественнонаучную сущность проблемы, формулировать задачи, определять пути их решения и оценивать эффективность выбора и методов правовой защиты результатов интеллектуальной деятельности с учетом исследований, разработки и проектирования биотехнических систем и технологий.

3 Модель

Решаемый пример соответствует определению усилий в цилиндре в бразильском тесте . Бразильский тест — это метод, который используется для косвенного измерения сопротивления горных пород. Это простой и эффективный метод, и поэтому он обычно используется для измерения горных пород. Иногда это испытание также используется для бетона [1] и он так же полезен в области биоматериалов для моделирования и исследования поведения материалов под нагрузкой в сочленениях: областей костей, связок и суставов.

На следующем рисунке представлена схема модели. Благодаря симметрии мы решаем задачу для сегмента. Для диска, который под нагрузкой может деформироваться в эллипс оправдано взять сегмент в 90 градусов. Соответственно, для моделирования сферы можно взять 90 градусов телесного угла, при этом алгоритм остается прежним.



4 Практическая часть - программирование в Jupyter Notebook

4.1 Подготовка окружения выполнения

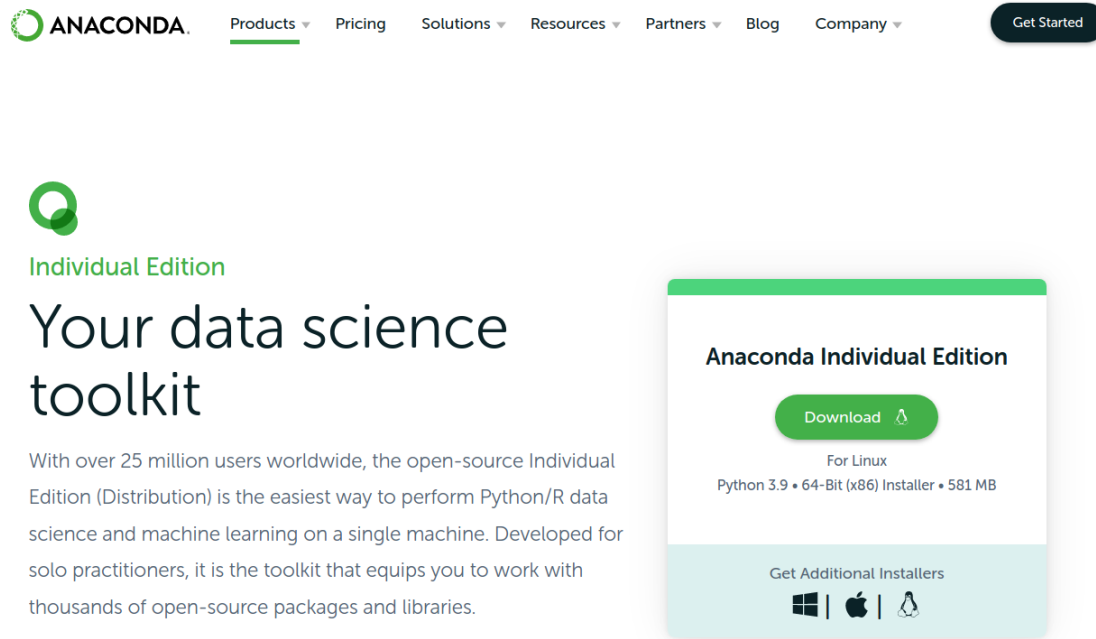
Для успешного выполнения кода на своем компьютере необходимо выполнить ряд подготовительных мероприятий.

4.1.1 Установка Anaconda

Anaconda - дистрибутив языков программирования Python и R, включающий набор популярных свободных библиотек, объединённых проблематиками науки о данных и машинного обучения.

Для скачивания пакета Anaconda надо перейти по ссылке и скачать дистрибутив для своей операционной системы:

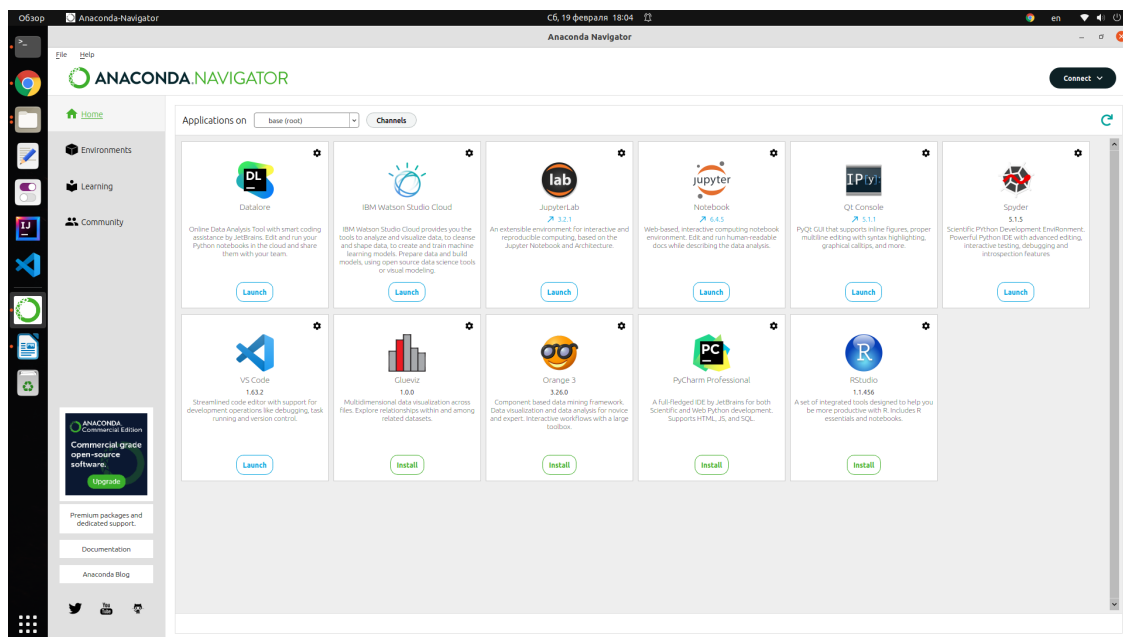
<https://www.anaconda.com/products/individual>



После установки необходимо запустить навигатор. В Windows и MacOS X его можно найти в списке приложений, в Linux его можно запустить из консоли

`./anaconda-navigator`

Окно навигатора показано на рисунке ниже.



4.1.2 Установка зависимостей

После установки Anaconda нам надо установить дополнительные библиотеки, для этого в консоли необходимо выполнить следующие команды. Их выполнение может занять значительное время, поэтому не стоит их прерывать из-за предположения, что они зависли. Если у вас возникнут сложности, поскольку со временем что то может измениться, то всегда можно обратиться с поиском в интернет и "загуглить" ошибку и способ ее устранения.

```
conda update --all --yes
conda install -c conda-forge google-colab
conda install -c conda-forge opencv
conda install -c conda-forge pandoc
conda install -c conda-forge nbconvert
conda install -c conda-forge rpy2
pip install solidspy
pip install easygui
pip install meshio
```

4.1.3 Установка Gmsh

Gmsh — это генератор 3D-сетки конечных элементов с открытым исходным кодом со встроенным движком САПР и постпроцессором. Цель его разработки — предоставить быстрый, легкий и удобный инструмент создания сетки с параметрическим вводом и расширенными возможностями визуализации. Gmsh построен вокруг четырех модулей: геометрия, сетка, решатель и постобработка.

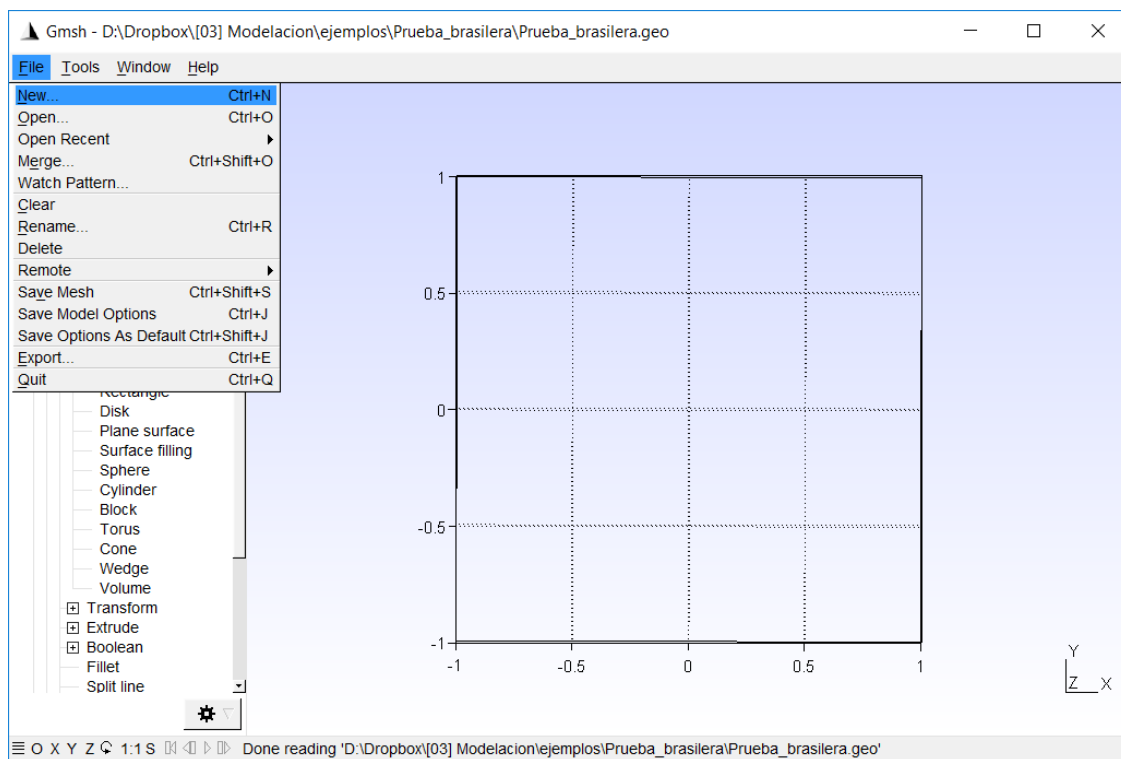
Для скачивания пакета Anaconda надо перейти по ссылке и скачать дистрибутив для своей операционной системы:

<https://gmsh.info/>

4.2 Пошаговая инструкция

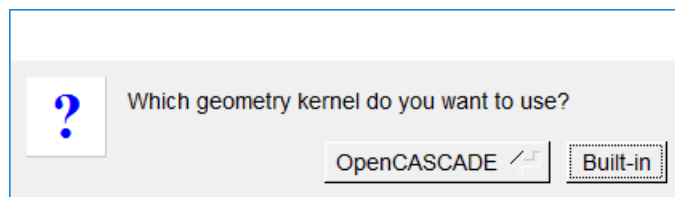
4.2.1 Создание геометрии в Gmsh

В качестве первого шага предлагается создать новый файл в Gmsh, как показано на следующем рисунке.

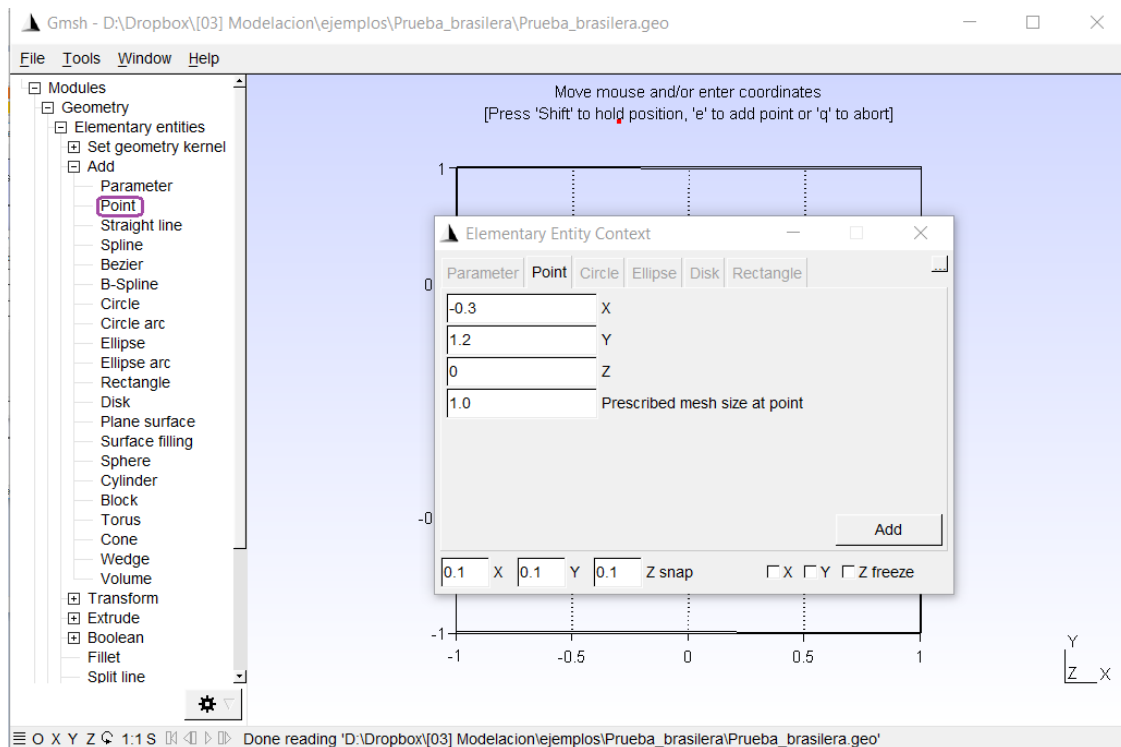


Если у вас на каком то шаге работы с Gmsh возникнут проблемы, то можете перейти к следующему разделу и создать файл в текстовом режиме. Gmsh это графический интерфейс, который позволяет сформировать текстовое описание модели, поэтому ничего страшного, если что-то у вас не получится, но если все пройдет в этом разделе "гладко то следующий раздел "Создание геометрии и сетки в текстовом режиме" можно пропустить.

При создании нового документа Gmsh может [1] задать вопрос о том, какое геометрическое ядро использовать. Не будем останавливаться на том, в чем отличия и воспользуемся built-in.



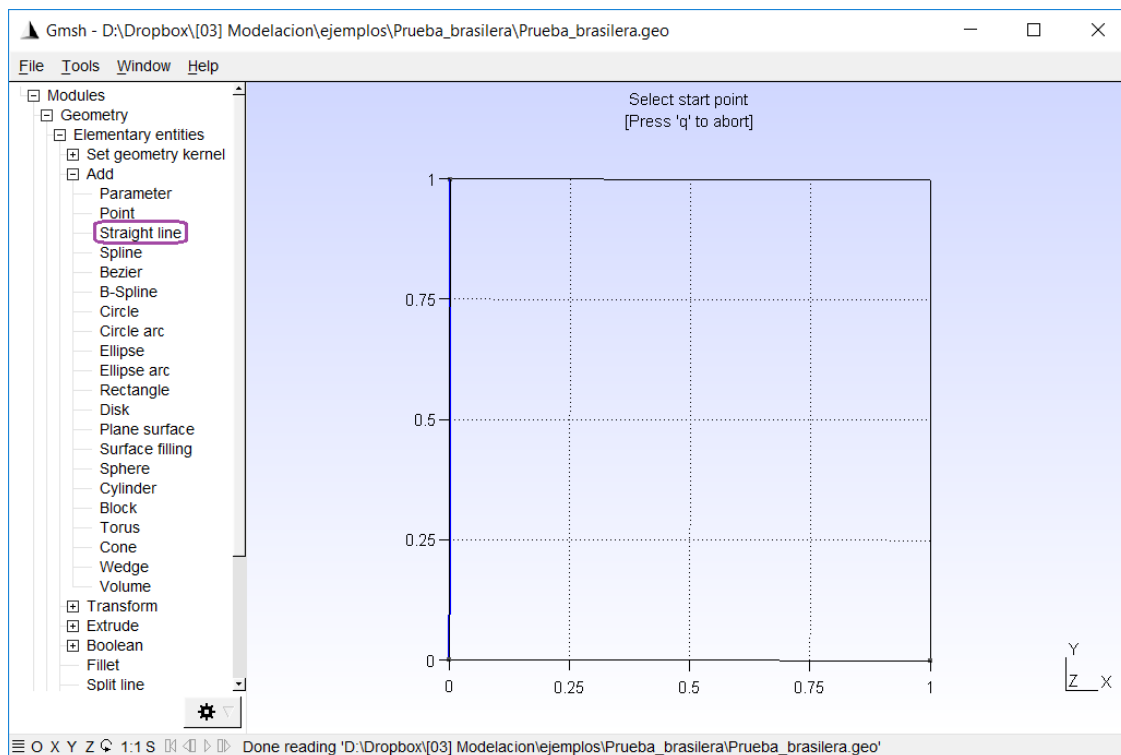
Для создания модели мы сначала создаем точки. Для этого перейдем к опции, как показано на следующем рисунке. Затем вводим координаты точек во всплывающем окне, нажимаем кнопку «Add» и, наконец, мы можем закрыть всплывающее окно.



На следующем шаге мы создаем линии. Для этого переходим к опции:

Geometry> Elementary Entities> Add> Straight line

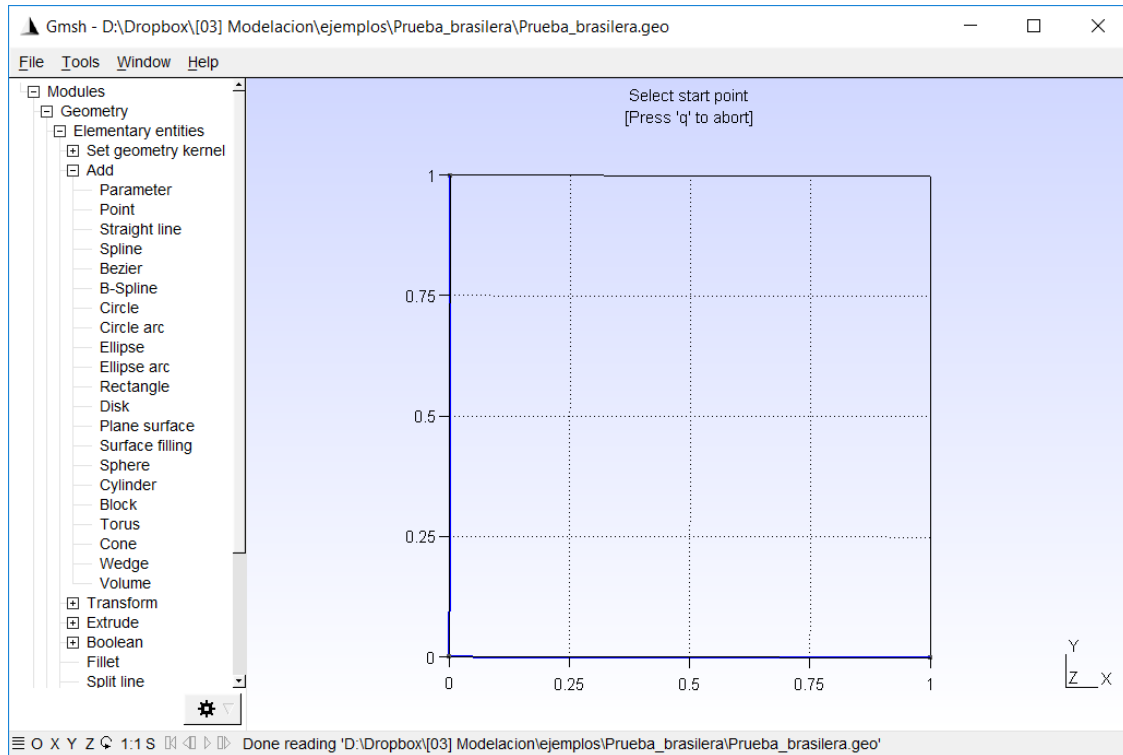
, как показано на следующем рисунке, и выбираем начальные точки и окончания для каждой линии. В конце нужно нажать "e" в английской раскладке.



Мы также создаем дуги окружности. Для этого переходим к опции:

Geometry > Elementary Entities > Add > Circle Arc

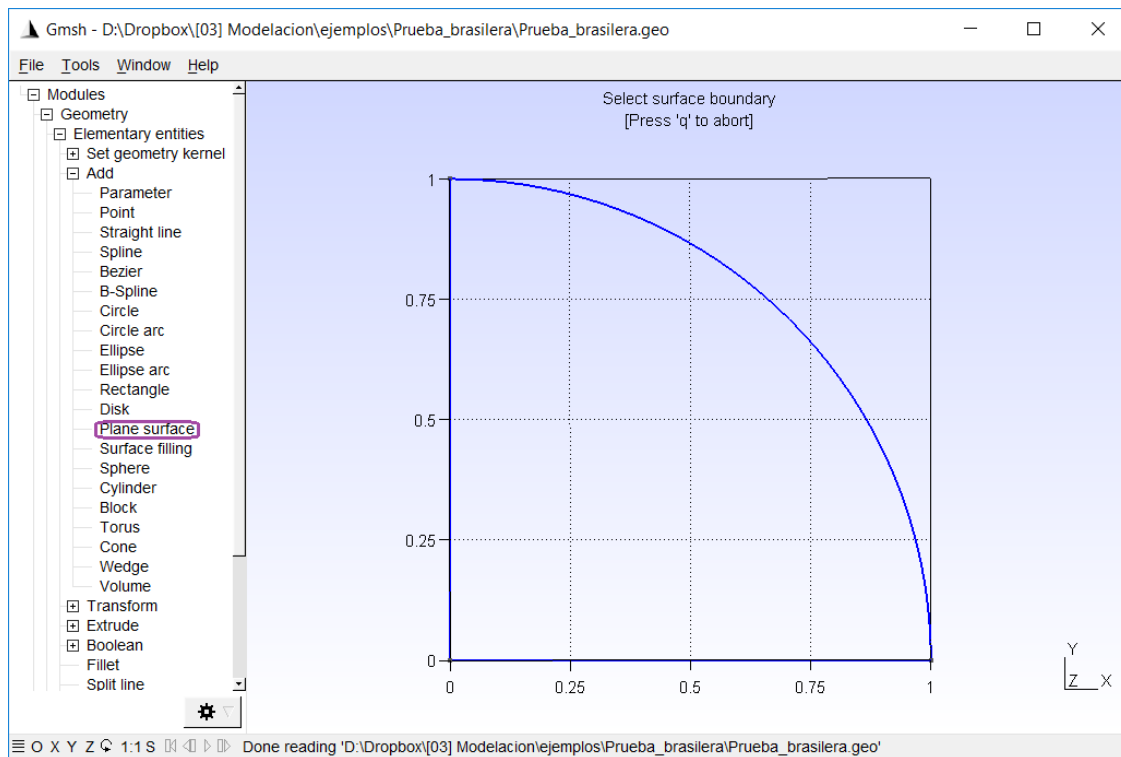
, как показано на следующем рисунке, и выбираем начальные точки, центральную и конечную для каждой дуги (в таком порядке). В конце нам нужно нажать "e" в английской раскладке.



Теперь, поскольку у нас уже есть замкнутый контур, мы можем определить поверхность. Для этого переходим к опции:

Geometry > Elementary Entities > Add > Plane Surface

, как показано на следующем рисунке, и выбираем контуры по порядку. В конце нам нужно нажать "e" в английской раскладке.



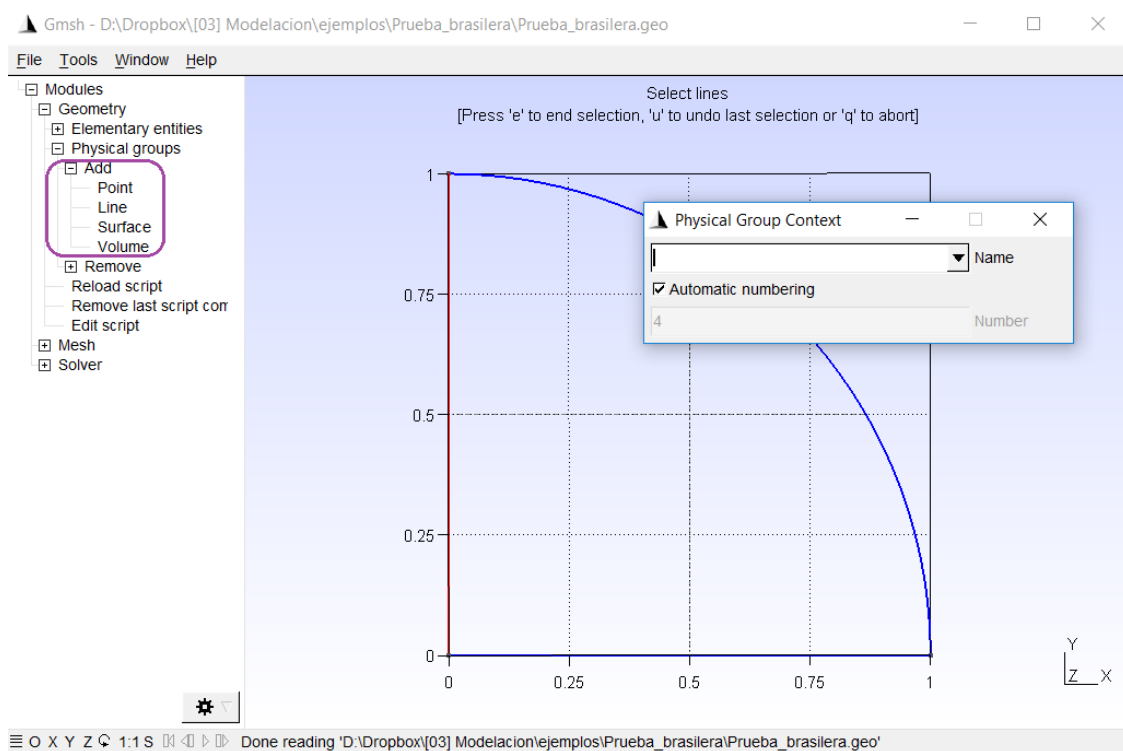
Теперь нам нужно определить физические группы. Физические группы позволяют нам связывать имена с различными частями модели, такими как линии и поверхности. Это позволит нам определить область, в которой мы будем разрешать модель (и свяжем материал), области с ограниченными движениями (граничные условия) и области, к которым мы применим нагрузку. В нашем случае у нас будет 4 физических группы:

- Область модели, где мы определим материал;
- Нижний край, где мы ограничим смещение по y ;
- Левый край, где мы ограничим смещение по x ;
- Верхняя точка, к которой мы будем прикладывать точечную нагрузку.

Чтобы определить физические группы, мы идем в

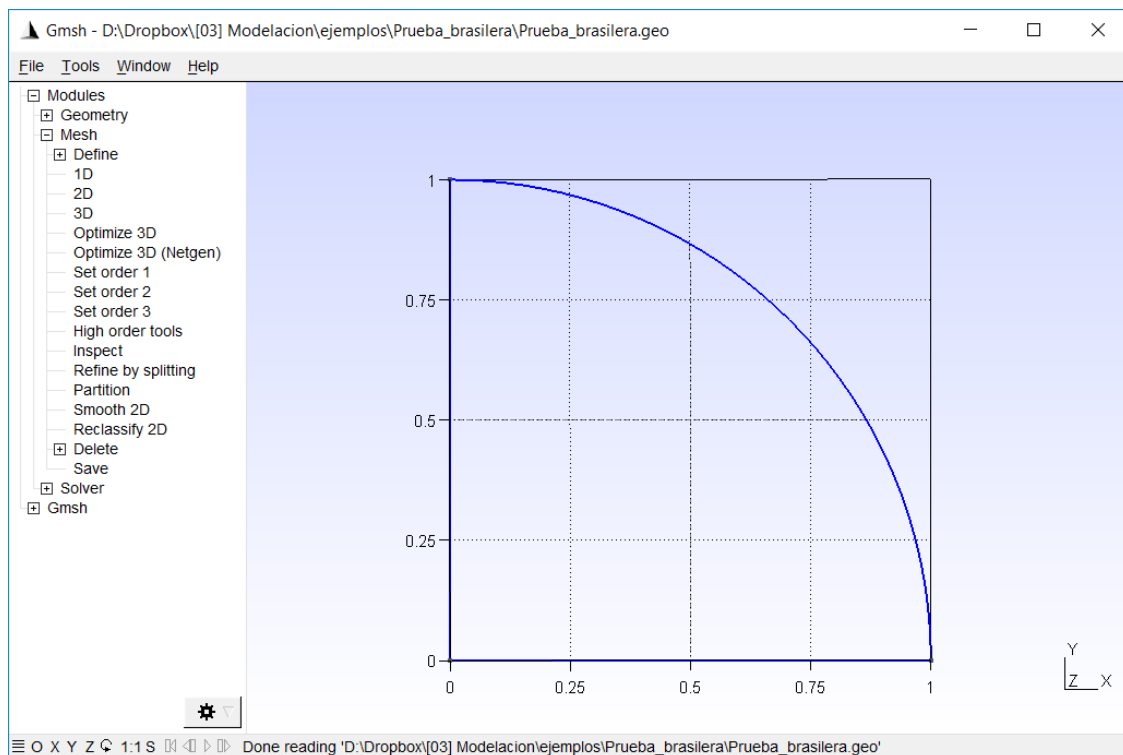
`Geometry> Physical groups> Add> Plane Surface`

, как показано на следующем рисунке. В этом случае мы можем оставить поле «Имя» пустым и позволить Gmsh назвать группы для нас, которые будут числами, которые мы затем можем просмотреть в текстовом файле.



4.2.2 Создание геометрии в текстовом режиме

Графическое представление файла (.geo) выглядит так:



Текстовый файл (.geo) выглядит так:

```

L = 0.1;

// Points
Point(1) = 0, 0, 0, L;
Point(2) = 1, 0, 0, L;
Point(3) = 0, 1, 0, L;

// Lines
Line(1) = 3, 1;
Line(2) = 1, 2;
Circle(3) = 2, 1, 3;

// Surfaces
Line Loop(1) = 2, 3, 1;
Plane Surface(1) = 1;

// Physical groups
Physical Line(1) = 1;
Physical Line(2) = 2;
Physical Point(3) = 3;
Physical Surface(4) = 1;

```

Вы можете проверить, что сохраненный файл соответствует листингу и в случае расхождения, поправить файл в любом текстовом редакторе.

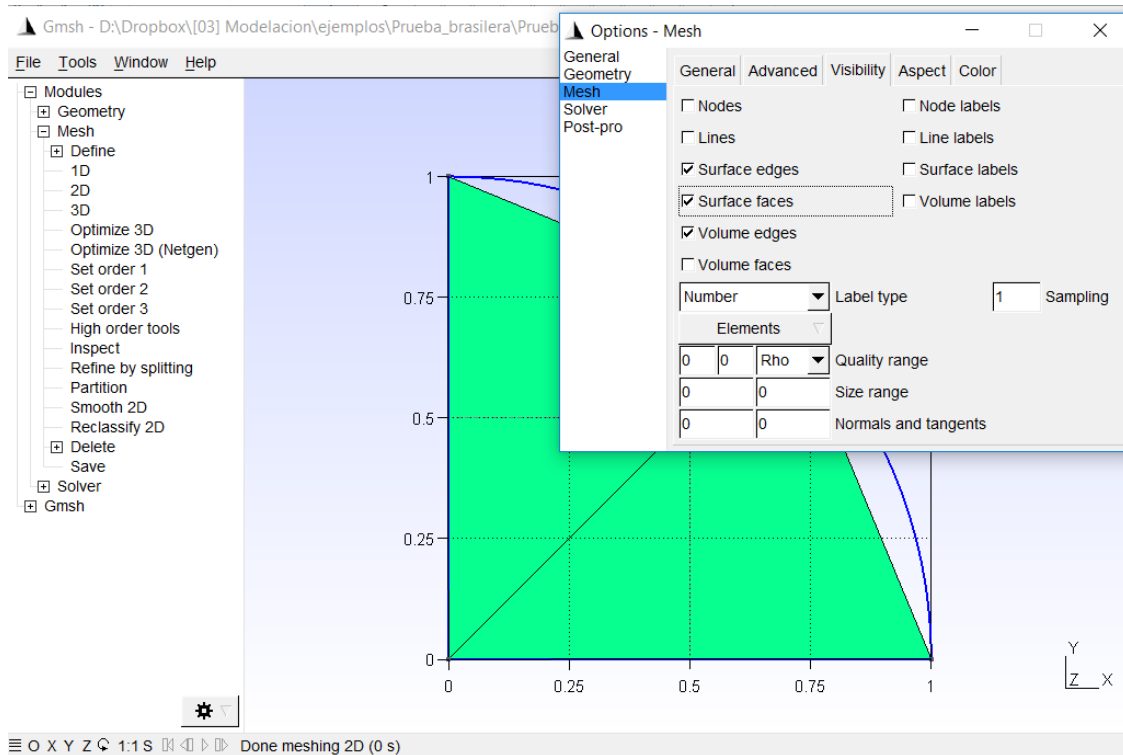
4.2.3 Создание сетки в Gmsh

Мы добавили параметр L , который мы можем изменить, чтобы изменить размер элементов при создании сетки.

Теперь приступаем к созданию сетки. Для этого переходим в

Mesh > 2D

,как мы видим на рисунке ниже:



4.2.4 Скрипт Python для создания входных файлов

Нам необходимо создать файлы с информацией об узлах (nodes.txt), элементах (eles.txt), нагрузках (loads.txt) и материалах (mater.txt).

Следующий код создает необходимые входные файлы для запуска программы конечных элементов в Python.

```
[1]: import matplotlib.pyplot as plt # load matplotlib
from solidspy import solids_GUI # import our package
disp = solids_GUI() # run the Finite Element Analysis
plt.show() # plot contours
```

Number of nodes: 9

Number of elements: 4

Number of equations: 14

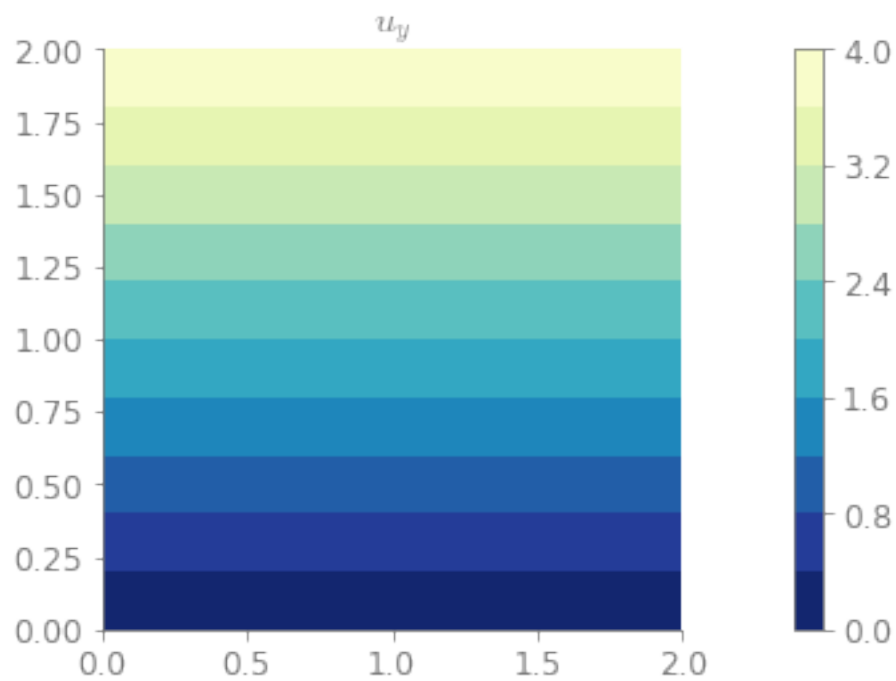
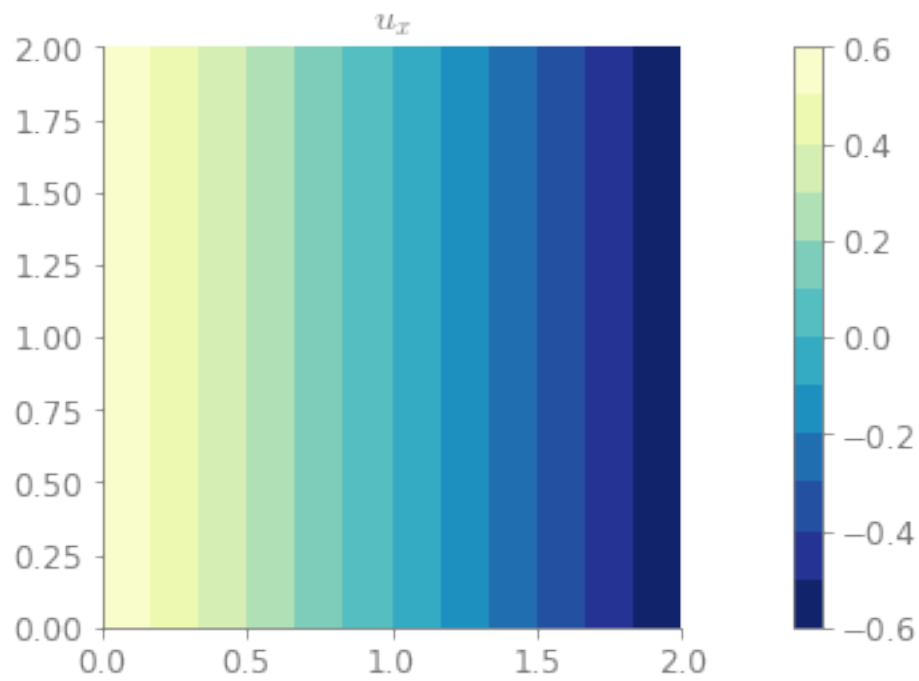
Duration for system solution: 0:00:00.080328

/home/il1/anaconda3/envs/intro_fem/lib/python3.9/site-packages/solidspy/postprocesor.py:108: UserWarning: The following kwargs were not used by contour: 'shading'

```
disp_plot(tri, field, levels, shading="gouraud")
```

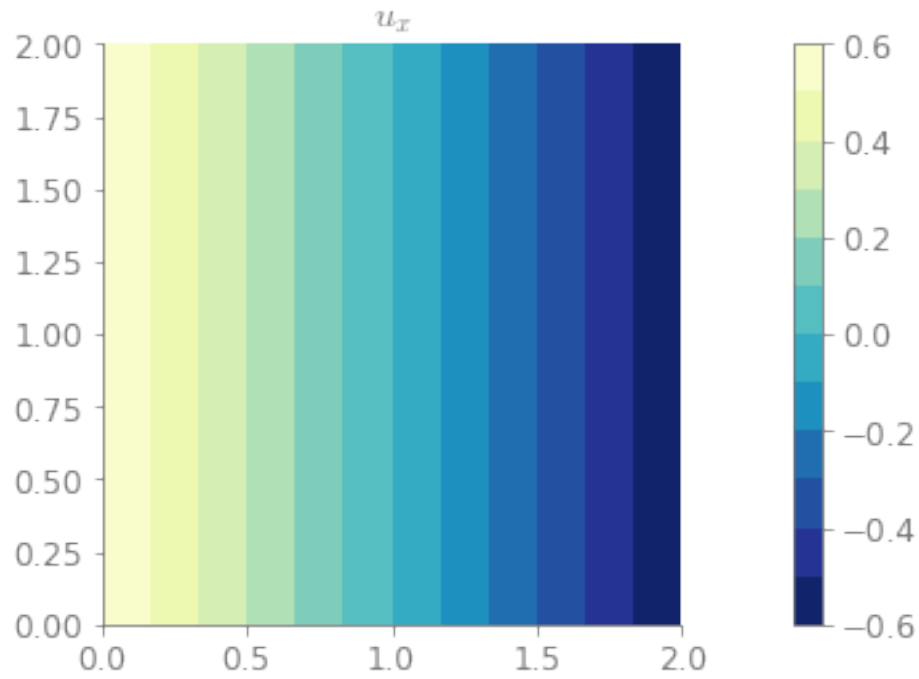
Duration for post processing: 0:00:00.191287

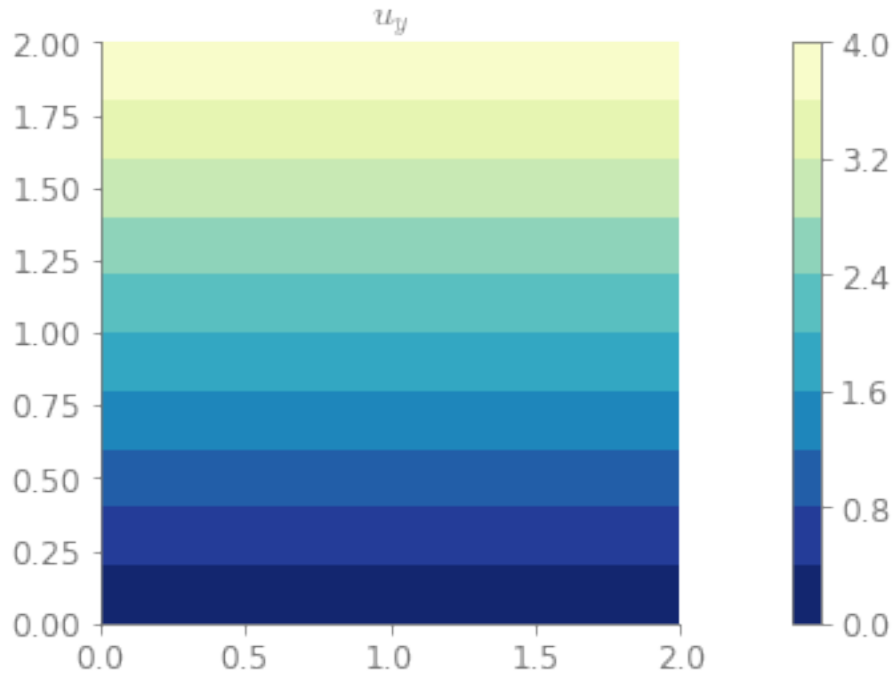
Analysis terminated successfully!



```
[2]: UC = solids_GUI()
```

Number of nodes: 9
Number of elements: 4
Number of equations: 14
Duration for system solution: 0:00:00.030502
Duration for post processing: 0:00:00.136590
Analysis terminated successfully!





```
[4]: import meshio
import numpy as np

mesh = meshio.read("g.msh")
points = mesh.points
cells = mesh.cells
point_data = mesh.point_data
cell_data = mesh.cell_data

# Element data
eles = cells["triangle"]
els_array = np.zeros([eles.shape[0], 6], dtype=int)
els_array[:, 0] = range(eles.shape[0])
els_array[:, 1] = 3
els_array[:, 3:] = eles

# Nodes
nodes_array = np.zeros([points.shape[0], 5])
nodes_array[:, 0] = range(points.shape[0])
nodes_array[:, 1:3] = points[:, :2]

# Boundaries
lines = cells["line"]
bounds = cell_data["line"]["gmsh:physical"]
```

```

nbounds = len(bounds)

# Loads
id_cargas = cells["vertex"]
nloads = len(id_cargas)
load = -10e8 # N/m
loads_array = np.zeros((nloads, 3))
loads_array[:, 0] = id_cargas
loads_array[:, 1] = 0
loads_array[:, 2] = load

# Boundary conditions
id_izq = [cont for cont in range(nbounds) if bounds[cont] == 1]
id_inf = [cont for cont in range(nbounds) if bounds[cont] == 2]
nodes_izq = lines[id_izq]
nodes_izq = nodes_izq.flatten()
nodes_inf = lines[id_inf]
nodes_inf = nodes_inf.flatten()
nodes_array[nodes_izq, 3] = -1
nodes_array[nodes_inf, 4] = -1

# Materials
mater_array = np.array([[70e9, 0.35],
                        [70e9, 0.35]])
maters = cell_data["triangle"]["gmsht:physical"]
els_array[:, 2] = [1 for mater in maters if mater == 4]

# Create files
np.savetxt("eles.txt", els_array, fmt="%d")
np.savetxt("nodes.txt", nodes_array,
           fmt=("%d", "%.4f", "%.4f", "%d", "%d"))
np.savetxt("loads.txt", loads_array, fmt=("%d", "%.6f", "%.6f"))
np.savetxt("mater.txt", mater_array, fmt="%.6f")

```

4.2.5 Визуализация модели

```

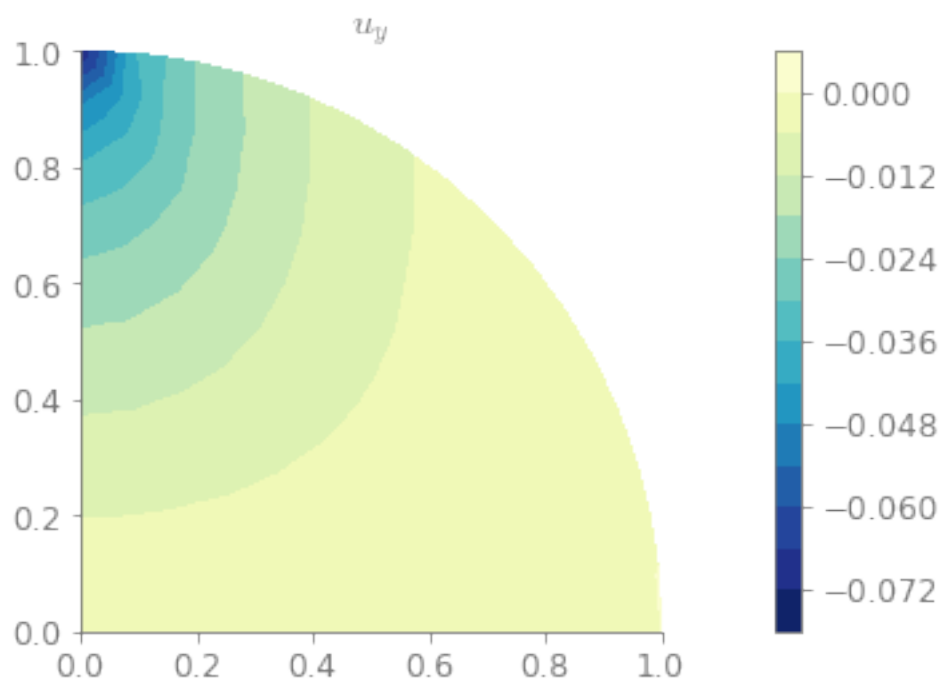
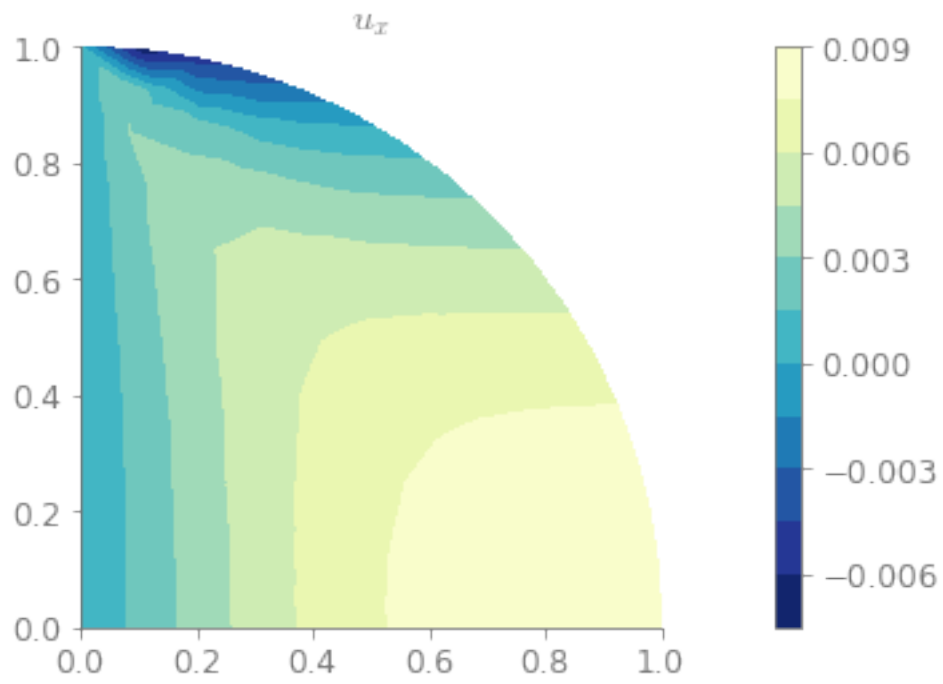
[5]: from solidspy import solids_GUI
     disp = solids_GUI()

```

```

Number of nodes: 119
Number of elements: 200
Number of equations: 216
Duration for system solution: 0:00:00.143867
Duration for post processing: 0:00:00.112697
Analysis terminated successfully!

```



[]:

5 Заключение

В результате практической работы был продемонстрирован на практике метод моделирования с помощью конечных элементов. Более подробную информацию можно почерпнуть из специальной литературы [2, 3, 4] и на сайте библиотеки <https://solidspy.readthedocs.io/en/latest/>.

Список литературы

- [1] ASTM D3967–16 (2016), Standard Test Method for Splitting Tensile Strength of Intact Rock Core Specimens, ASTM International, www.astm.org.
- [2] Geuzaine, Christophe, y Jean-François Remacle (2009), Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities. International Journal for Numerical Methods in Engineering, 79.11.
- [3] Geuzaine, Christophe, y Jean-François Remacle (2017), Gmsh Official Tutorial. <http://gmsh.info/doc/texinfo/gmsh.html>.
- [4] Geuzaine, Christophe, y Jean-François Remacle (2017), Gmsh Official Screencasts. <http://gmsh.info/screencasts/>.