

Отчёт о модификации проекта «Изображение проекции полиэдра»

И.Д. Соловьев Е.А. Роганов А.И. Александров

26.04.2024

Содержание

Постановка задачи	1
Точная постановка задачи	1
Интерпретация задачи и идея решения	1
Подробное описание решения задачи на модификацию	2
Команды, использовавшиеся при получении итоговых отчетов в заданных форматах	2
Примеры тестов для подтверждения корректности работы модификации	3

*

Постановка задачи

Точная постановка задачи

Назовём точку в пространстве «хорошей», если её проекция находится строго внутри окружности $x^2 + y^2 = 4$, но строго вне окружности $x^2 + y^2 = 1$. Модифицируйте эталонный проект таким образом чтобы определялась и печаталась следующая характеристика полиэдра: сумма длин рёбер, оба из концов которых — «хорошие» точки.

Интерпретация задачи и идея решения

В данной задаче требуется рассматривать проекции вершин полиэдра на плоскость Oxy , и складывать длины таких ребер полиэдра, оба конца которых попадают в заданное кольцо, то есть удовлетворяют неравенству:

$$1 \leq M_x^2 + M_y^2 \leq 4,$$

где M — рассматриваемая точка. Также необходимо реализовать графический интерфейс, отображающий заданное кольцо для визуального контроля правильности работы модификации.

Подробное описание решения задачи на модификацию

1. В класс `R3` был добавлен метод, проверяющий, является ли переданная точка «хорошей», при помощи подстановки ее x и y координат в рассмотренное при постановке задачи неравенство.
2. Было принято решение выявлять ребра, оканчивающиеся «хорошими» точками на этапе задания ребер для каждой грани в методе `__init__` класса `Polyedr`.
3. Так как некоторые ребра могут принадлежать одной или двум граням, в зависимости от конфигурации полиэдра, то необходимо проверять, было ли ранее посчитано рассматриваемое ребро, или нет. Самой простой с точки зрения написания кода проверкой в данном случае является двумерный массив смежности. Это «таблица», по столбцам и строкам которой расположены все вершины полиэдра, ячейки заполнены единицами. Когда вычисляется сумма очередного ребра, ячейка на пересечении строки и столбца с соответствующими вершинами начала и конца ребра заполняется нулем.
4. Также в метод `__init__` класса `Polyedr` был добавлен дополнительный массив вершин, с непреобразованными координатами, чтобы в полученной суммарной длине не учитывался коэффициент гомотетии.
5. Вычисление длины ребер было реализовно с помощью операции взятия квадратного корня из скалярного произведения.
6. Была добавлена отправка в стандартный вывод информации о длине ребер, подходящих под условие.
7. В класс `Polyedr` добавляется метод `print_good_sum(self)`, возвращающий искомую сумму длин ребер с «хорошими» точками на концах.
8. В методе `draw` класса `Polyedr` была добавлена отрисовка кольца и проекций точек, при этом учтен коэффициент гомотетии для корректного отображения кольца.
9. Были написаны соответствующие тесты, иллюстрирующие правильность работы всех написанных методов и модификации в целом.

Команды, использовавшиеся при получении итоговых отчетов в заданных форматах

Здесь `pdf.latex` и `HTML.html5` — переименованные файлы шаблонов.

Команда для получения отчета в формате `pdf`, с использованием заданного шаблона.

```
pandoc --template pdf.latex -s --toc --lua-filter ./include-code-files.lua report.md -o report.pdf
```

Команда для получения отчета в формате `html`, с использованием заданного шаблона.

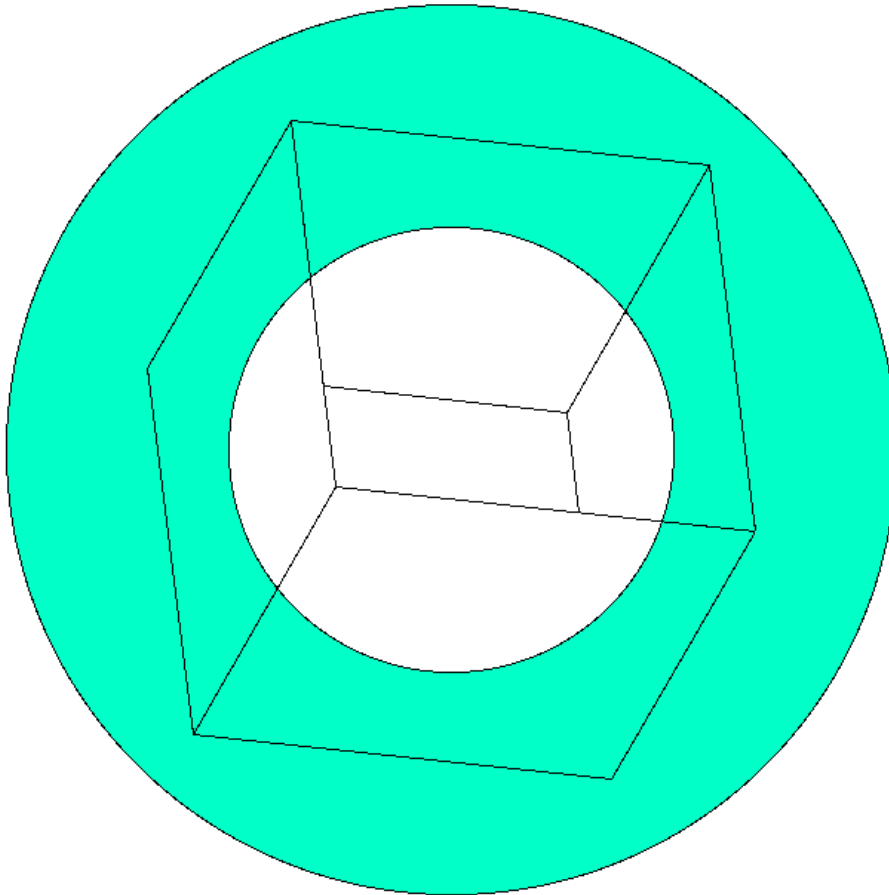
```
pandoc -o report.html -f markdown -t html -s --toc --lua-filter ./include-code-files.lua --mathjax
```

Команда для получения отчета в формате `docx`.

```
pandoc -s report.md --toc --lua-filter ./include-code-files.lua -o report.docx
```

Примеры тестов для подтверждения корректности работы модификации

- Коробка, шесть из восьми вершин которой попадают в заданное кольцо.



```
@classmethod
def setUpClass(self):
    fake_file_content = """200.0      60.0      -140.0      60.0
8      5      20
-1      -1      1
-1      1      1
1      1      1
1      -1      1
-1      -1      -1
-1      1      -1
1      1      -1
1      -1      -1
4      1      2      3      4
4      5      6      2      1
4      3      2      6      7
4      3      7      8      4
4      1      4      8      5"""
```

```

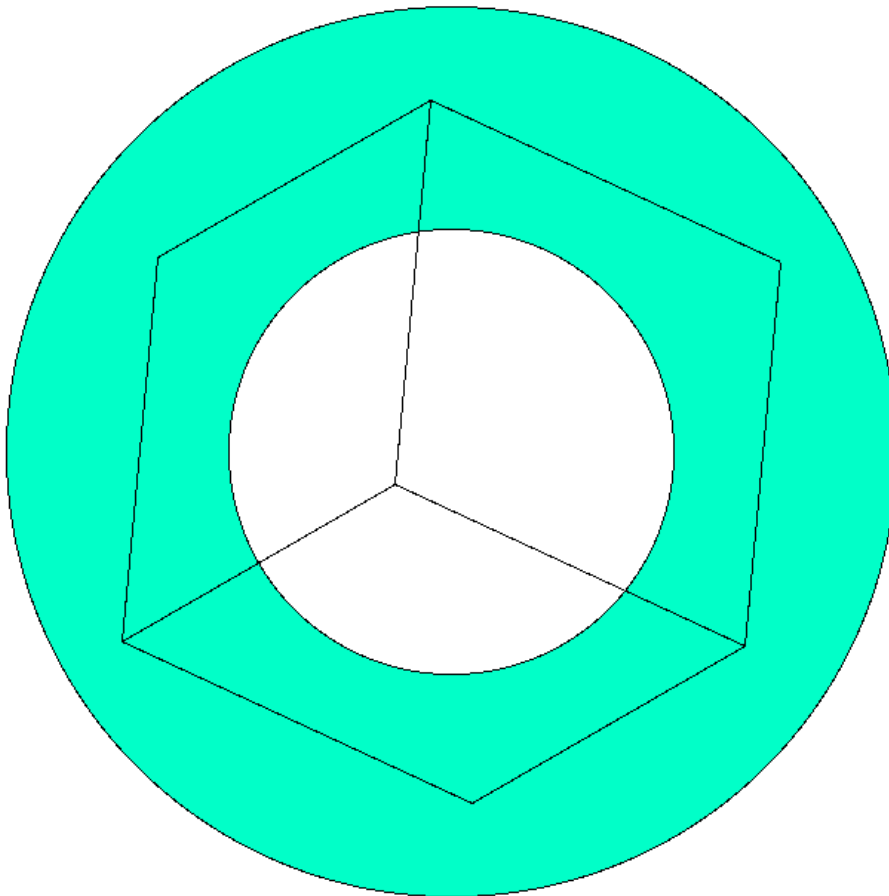
fake_file_path = 'data/poly_box.geom'
with patch('shadow.polyedr.open'.format(__name__),
          new=mock_open(read_data=fake_file_content)) as _file:
    self.poly_box = Polyedr(fake_file_path)
    _file.assert_called_once_with(fake_file_path)

# polyedr perimetr

def test_box_perimetr(self):
    self.assertEqual(self.poly_box.good_perimetr, 12)

```

- Куб, шесть из восьми вершин которого попадают в заданное кольцо.



```

@classmethod
def setUpClass(self):
    fake_file_content = """200.0      45.0      45.0      30.0
8      6      24
-1      -1      1
-1      1      1
1      1      1
1      -1      1

```

```

-1      -1      -1
-1      1      -1
1       1      -1
1      -1      -1
4       1      2      3      4
4       5      6      2      1
4       3      2      6      7
4       3      7      8      4
4       1      4      8      5
4       8      7      6      5"""
fake_file_path = 'data/poly_cube.geom'
with patch('shadow.polyedr.open'.format(__name__),
           new=mock_open(read_data=fake_file_content)) as _file:
    self.poly_cube = Polyedr(fake_file_path)
    _file.assert_called_once_with(fake_file_path)

# double edges (with two facets) can be countable

def test_cube_perimetr(self):
    self.assertEqual(self.poly_cube.good_perimetr, 12)

```