

**Московский авиационный институт  
(Национальный исследовательский университет)**

Факультет: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Компьютерная графика»

**Лабораторная работа № 3**

**Тема: Основы построения фотореалистичных изображений.**

Студентка: Довженко А.А.

Группа: 80-307

Преподаватель: Чернышов Л.Н.

Дата:

Оценка:

Москва, 2018

**1. Постановка задачи**

Используя результаты Л.Р.№2, аппроксимировать заданное тело выпуклым многогранником. Точность аппроксимации задается

пользователем. Обеспечить возможность вращения и масштабирования многогранника и удаление невидимых линий и поверхностей. Реализовать простую модель закраски для случая одного источника света.

Параметры освещения и отражающие свойства материала задаются пользователем в диалоговом режиме.

Вариант 7.

Одна из полостей двуполостного гиперboloида.

## 2. Решения задачи

Для решения задачи я использовала библиотеку `plotly`, которая позволяет выполнить поставленную задачу. Я строю `trisurf` график, аппроксимирую гиперboloид треугольниками. Для реализации модели освещения использую встроенные параметры `ambient` (окружающее освещение), `diffuse` (рассеянное освещение), `fresnel` (коэффициент отражения и пропускания света), `specular` (способность отражения материала), `roughness` (шероховатость поверхности).

## 3. Руководство по использованию программы

Запустить `jupyter notebook` и в браузере выбрать тетрадь с программным кодом. Поочередно выполнить все ячейки в тетради.

## Запуск jupyter notebook из командной строки.

```
C:\WINDOWS\system32\cmd.exe - jupyter notebook
Microsoft Windows [Version 10.0.17134.407]
(c) Корпорация Майкрософт (Microsoft Corporation), 2018. Все права защищены.

C:\Users\Karma>cd Desktop
C:\Users\Karma\Desktop>cd cg
C:\Users\Karma\Desktop\cg>cd lab3
Системе не удастся найти указанный путь.
C:\Users\Karma\Desktop\cg>cd lab3
C:\Users\Karma\Desktop\cg\lab3>jupyter notebook
[I 04:58:55.333 NotebookApp] The port 8888 is already in use, trying another port.
[I 04:58:55.668 NotebookApp] Serving notebooks from local directory: C:\Users\Karma\Desktop\cg\lab3
[I 04:58:55.668 NotebookApp] 0 active kernels
[I 04:58:55.670 NotebookApp] The Jupyter Notebook is running at: http://localhost:8889/?token=52b44c97f39483cb61ad0d1d56c1adc2a095d5a5469ef410
[I 04:58:55.670 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 04:58:55.680 NotebookApp]

Copy/paste this URL into your browser when you connect for the first time,
to login with a token:
http://localhost:8889/?token=52b44c97f39483cb61ad0d1d56c1adc2a095d5a5469ef410
[I 04:58:56.021 NotebookApp] Accepting one-time-token-authenticated connection from ::1
[W 04:58:56.837 NotebookApp] 404 GET /favicon.ico (::1) 97.60ms referer=None
```

## Описание лабораторной работы.

**Лабораторная работа №3**

**Тема: Основы построения фотореалистичных изображений.**

**Довженко Анастасия М8О-307Б**

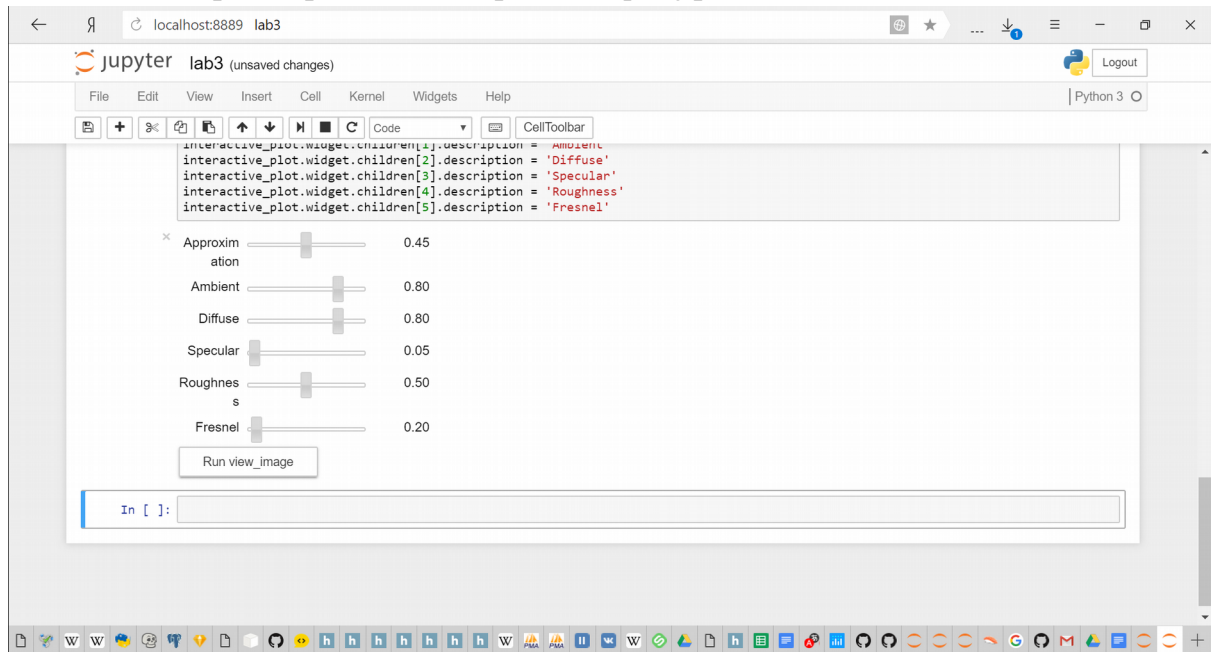
**Вариант: Одна из полостей двуполостного гиперболоида.**

Задание: Используя результаты Л.Р.№2, аппроксимировать заданное тело выпуклым многогранником. Точность аппроксимации задается пользователем. Обеспечить возможность вращения и масштабирования многогранника и удаление невидимых линий и поверхностей. Реализовать простую модель закраски для случая одного источника света.

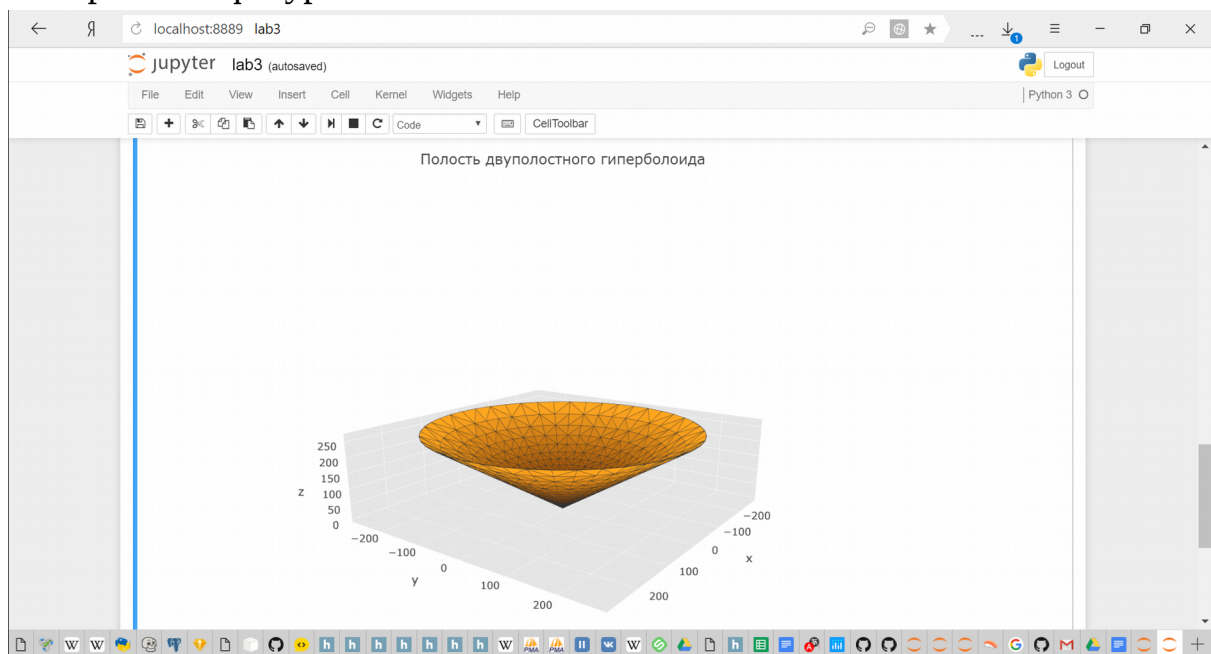
```
In [3]: interactive_plot = interact_manual(view_image, appr=(0.1, 0.8, 0.05),
                                         a=(0.0, 1.0, 0.1),
                                         d=(0.0, 1.0, 0.1),
                                         s=(0.0, 2.0, 0.05),
                                         r=(0.0, 1.0, 0.1),
                                         f=(0.0, 5.0, 0.2))

interactive_plot.widget.children[0].description = 'Approximation'
interactive_plot.widget.children[1].description = 'Ambient'
interactive_plot.widget.children[2].description = 'Diffuse'
interactive_plot.widget.children[3].description = 'Specular'
interactive_plot.widget.children[4].description = 'Roughness'
interactive_plot.widget.children[5].description = 'Fresnel'
```

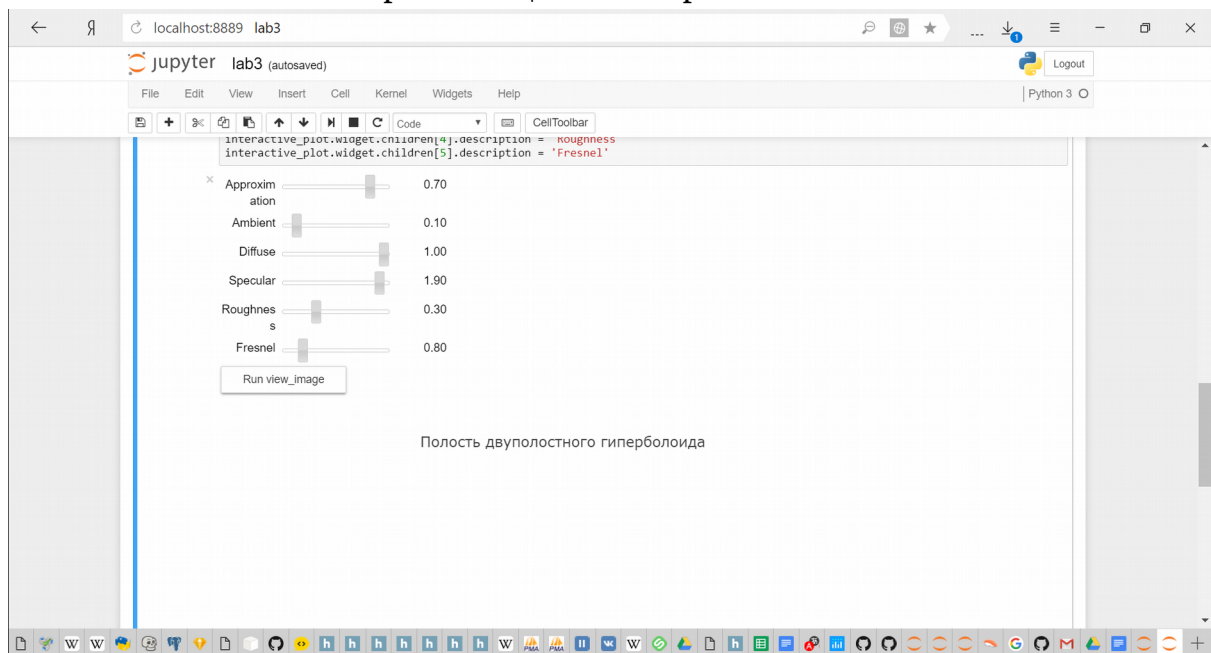
## Исходные параметры для построения фигуры.



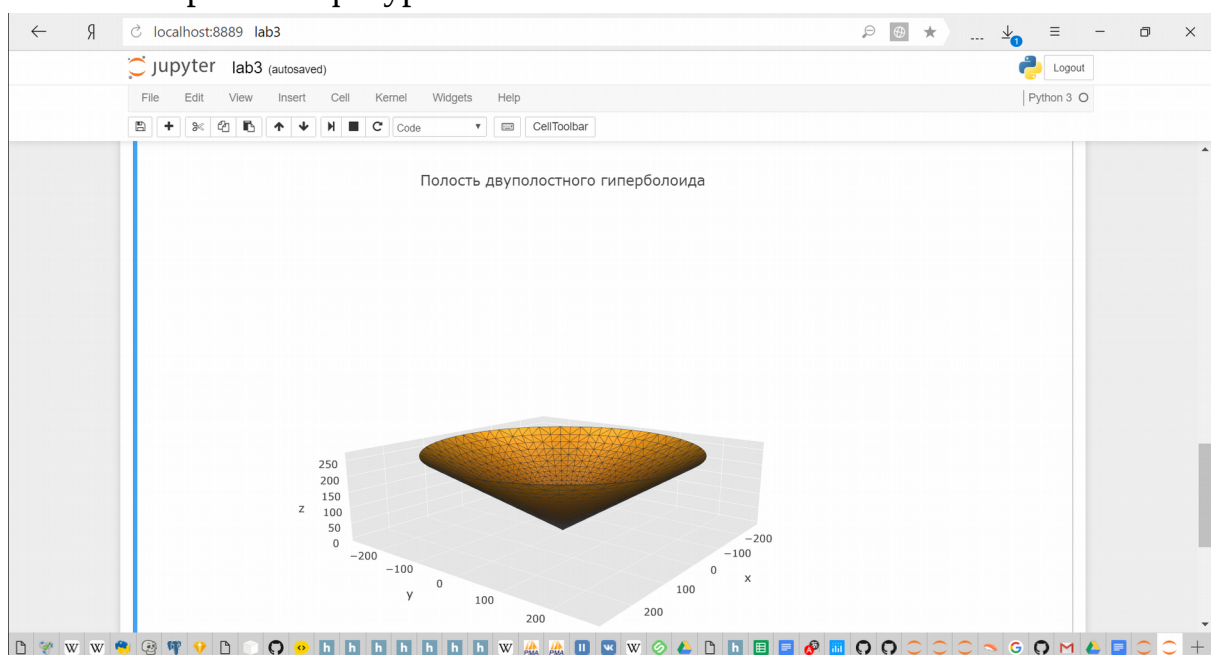
## Построенная фигура.



Изменяем точность аппроксимации и настройки света.



Новая построенная фигура.



#### 4. Листинг программы'

# Лабораторная работа №3

### Тема: Основы построения фотореалистичных изображений.

### Довженко Анастасия М8О-307Б

### Вариант: Одна из полостей двуполостного гиперboloида.

Задание: Используя результаты Л.Р.№2, аппроксимировать заданное тело выпуклым многогранником. Точность аппроксимации задается пользователем. Обеспечить возможность вращения и масштабирования многогранника и удаление невидимых линий и поверхностей. Реализовать простую модель закраски для случая одного источника света.

```
import plotly
import plotly.plotly as py
import plotly.figure_factory as FF
from plotly.graph_objs import graph_objs as go
from IPython.html.widgets import interact, interact_manual, Layout, HBox,
VBox
```

```
import numpy as np
from scipy.spatial import Delaunay
```

```
plotly.offline.init_notebook_mode()
from plotly.offline import iplot
```

```
def view_image(appr, a=0.8, d=0.8, s=0.05, r=0.5, f=0.2):
```

```
    u = np.linspace(0, 2*np.pi, int(appr*100))
    v = np.linspace(0, 2*np.pi, int(appr*100))
    u,v = np.meshgrid(u,v)
    u = u.flatten()
    v = v.flatten()
```

```
    x = np.cosh(u)*np.cos(v)
    y = np.cosh(u)*np.sin(v)
    z = np.sinh(u)
```

```
    points2D = np.vstack([u,v]).T
    tri = Delaunay(points2D)
    simplices = tri.simplices
```

```
    def dist_origin(x, y, z):
```

```

return np.sqrt((1.0 * x)**2 + (1.0 * y)**2 + (1.0 * z)**2)

hyperboloid = FF.create_trisurf(x=x, y=y, z=z, color_func=dist_origin,
                               colormap = [(0.4, 0.15, 0), (1, 0.65, 0.12)],
                               show_colorbar=False,
                               simplices=simplices, title="Полость двуполостного
гиперboloида",
                               aspectratio=dict(x=1, y=1, z=0.3))
hyperboloid['data'][0].update(lightning=dict(ambient=a,
                                              diffuse=d,
                                              fresnel=f,
                                              specular=s,
                                              roughness=r,
                                              facenormalsepsilon=1e-15,
                                              vertexnormalsepsilon=1e-15),
                              lightposition=dict(x=0, y=0, z=400)
                              )
iplot(hyperboloid)

interactive_plot = interact_manual(view_image, appr=(0.1, 0.8, 0.05),
                                   a=(0.0, 1.0, 0.1),
                                   d=(0.0, 1.0, 0.1),
                                   s=(0.0, 2.0, 0.05),
                                   r=(0.0, 1.0, 0.1),
                                   f=(0.0, 5.0, 0.2))

```

```

interactive_plot.widget.children[0].description = 'Approximation'
interactive_plot.widget.children[1].description = 'Ambient'
interactive_plot.widget.children[2].description = 'Diffuse'
interactive_plot.widget.children[3].description = 'Specular'
interactive_plot.widget.children[4].description = 'Roughness'
interactive_plot.widget.children[5].description = 'Fresnel'

```

## 5. Выводы

Я научилась аппроксимировать тело, освежила и углубила свои знания по линейной алгебре. Узнала о разных моделях освещения и их

программной реализации. Библиотека plotly имеет широкий спектр применений: как построение простейших графиков, так и сложных 3D моделей. Жаль, что мне не удалось попробовать больше возможностей этой библиотеки, надеюсь, это удастся сделать в будущем.

#### 6. Используемые источники

1. Документация библиотеки plotly [Электронный ресурс]. URL: <https://plot.ly/python/> (дата обращения: 22.11.2018).
2. Исходный код библиотеки plotly [Электронный ресурс]. URL: <https://github.com/plotly/plotly.py> (дата обращения: 24.11.2018).
3. Документация Jupyter Widgets [Электронный ресурс]. URL: <https://ipywidgets.readthedocs.io/en/stable/index.html> (дата обращения: 24.11.2018).