МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский университет ИТМО»
(Университет ИТМО)

Факультет систем управления и робототехники

ПРОЕКТ
по дисциплине
«Механика (продвинутый уровень)»

по теме:

ДВИЖЕНИЕ СТРАТОСТАТА С УЧЕТОМ ИЗМЕНЕНИЯ ПЛОТНОСТИ ВОЗДУХА С ВЫСОТОЙ ПРИ ВЗЛЕТЕ С ПОВЕРХНОСТИ ЗЕМЛИ

Работу выполнили:
Д.А. Возжаева Группа R3135
Д.Р. Алиев Группа R3135

Предподаватель:
Старший преподаватель, физический факультет *А.В. Смирнов*

Санкт-Петербург 2023

СОДЕРЖАНИЕ

1 ВВЕДЕНИЕ

Моделирование полета стратостата - это область науки, которая привлекает к себе внимание ученых и инженеров со всего мира. Стратостаты, или воздушные шары на высоте, являются важным инструментом для проведения научных и коммерческих исследований на высотах до 50 км от уровня моря. Эти устройства позволяют тщательно исследовать атмосферные явления, космическое излучение и химические процессы, которые происходят на границе между атмосферой и космосом.

Мы поставили перед собой цель разработать компьютерное приложение с удобным интерфейсом, симулирующее полет стратостата в зависимости от введенного значения его плотности. Для достижения поставленной цели необходимо сделать следующее:

1. Изучить строение летательного аппарата и физические законы, которым он подчиняется;
2. Составить математическую модель стратостата;
3. Решить полученные дифференциальные уравнения;
4. Написать код на языке Python на основе полученных алгоритмов
5. Подобрать подходящую библеотеку, разработать и реализовать интерфейс.

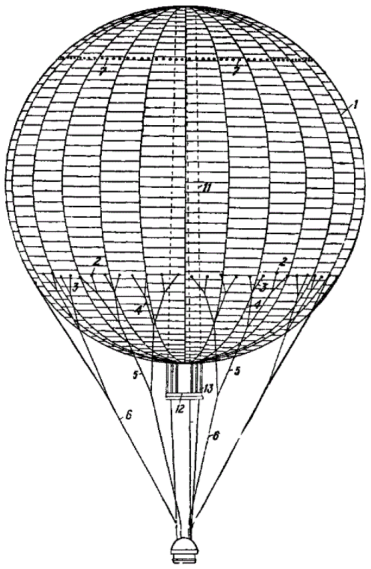
2 ФИЗИЧЕСКОЕ МОДЕЛИРОВАНИЕ

2.1 Строение летательного аппарата

Стратостат — это устройство, предназначенное для выполнения научных и коммерческих задач на высотах от 20 до 50 км от уровня моря. Его строение наиболее оптимально при этом строится точечно в соответствии с задачами, которые нужно решить. В зависимости от целей полета можно использовать разное оборудование и определенные элементы конструкции.

Одним из основных элементов строения стратостата является оболочка или тело, которое обеспечивает сохранение воздушного шара в воздухе. Оболочка может быть изготовлена из различных материалов, таких как легкие металлы и пластик, и формы, в которой она может иметь, могут быть самыми разными. На оболочке могут быть установлены окна, через которые идет наблюдение за окружающей средой и происходит сбор данных.

Подвеска - это механизм, с помощью которого на стратостате закрепляется научное оборудование и другое оборудование, необходимое для проведения исследований. Она может быть выполнена из металлических конструкций и необходима для обеспечения устойчивости оборудования в полете.



Гелиевые баллоны - это основной источник поддержания воздушного шара в воздухе. Гелий имеет меньшую плотность, поэтому обеспечивает необходимую подъемную силу, которая позволяет свободно перемещаться по воздуху. Размер гелиевых баллонов, количество их и местоположение на стратостате зависят от размера и целей полета.

Система управления стратостатом служит для регулировки высоты, направления и скорости полета. Она включает в себя компьютеры, большое количество сенсоров и другие устройства, необходимые для контроля и управления полетом стратостата.

Таким образом, строение стратостата определяется целями его использования и наиболее оптимально строится под необходимые задачи.

2.2 Математическая модель

Рассмотрим полет стратостата с поверхности земли, то есть в начальный момент времени t=0 высота подъема равна нулю. При построении модели учтем сопротивление воздуха во время подъема, изменение плотности воздуха на разной выосте, Архимедову силу, действующую на стратостат. Получаем слудующую формулу:

$$\frac{d^2y}{dt^2} = g\frac{\rho_a - \rho_s}{\rho_s} - k\frac{dy}{dt}$$

Произведем замену $\frac{dy}{dt} = V$:

$$\frac{dV}{dt} = g\frac{\rho_a - \rho_s}{\rho_s} - kV$$

Разнесем дифференциала по обе стороны от знака раенства:

$$\frac{dV}{g(\rho_a - \rho_s) - kV\rho_s} = \frac{dt}{\rho s}$$

Поскольку dt очень мало, значит и изменение координаты y тоже не велико, поэтому примем ρ_a константой на промежутке интегрировния:

$$\int_{V_0}^{V} \frac{dV}{g(\rho_a - \rho_s) - kV\rho_s} = \int_0^{\Delta t} \frac{dt}{\rho_s}$$

Вычислим собственный интеграл:

$$\ln|\frac{g(\rho_a - \rho_s) - kV\rho_s}{g(\rho_a - \rho_s) - kV_0\rho_s}| = -k\Delta t$$

Потенциируем равенство, при этом заметим, что правая часть выражения положительна, а значит можем снять модуль:

$$\frac{g(\rho_a - \rho_s) - kV\rho_s}{g(\rho_a - \rho_s) - kV_0\rho_s} = e^{-k\Delta t}$$

Выразим зависимость скорости от времени:

$$V(\Delta t) = \frac{g(\rho_a - \rho_s)(1 - e^{-k\Delta t}) + kV_0\rho_s e^{-k\Delta t}}{k\rho_s}$$

2.3 Плотность воздуха

Очевидно, на разной высоте воздух имеет различную плтоность. Состояние атмосферы Земли можно описать следующим уравнением гидростатики:

$$\vec{\nabla} P = \rho \vec{F}$$

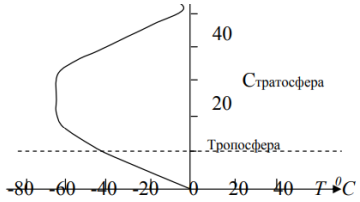
Это уравнение может быть получено из уравнений Эйлера для идеальной жидкости при отсутствии движения. В поле сил тяжести уравнение гидростатики примет вид:

$$\frac{dP}{dy} = -\rho g$$

Будем считать атмосеру идеальным газом, тогда можем записать уравнение Клайперона-Менделеева:

$$\frac{dP}{dy} = -\rho g$$

Исследуем полет стратостата на высоту до 35км.



Как видно на схеме, температура воздуха падает до высоты 20км, а далее остается постоянной. Примем зависимость плотности воздуха равной:

$$\rho_a(y) = \begin{cases} \rho_0 10^{-\beta \frac{y}{1000}} & y < 20000 \\ \rho_0 10^{-20\beta} & y \geq 20000 \end{cases}$$

Где ρ_0 - плотность воздуха на уровне моря, $\beta = 0.125$

3 КОМПЬТЕРНОЕ МОДЕЛИРОВАНИЕ

3.1 Выбор библиотеки

Далее нам необходимо было подобрать удобную библиотеку для написания нашего приложения. Нам удалось найти три наиболее подходящих библиотеки: Matplotlib, Dearpygui и Manim. Для нас было важно, чтобы библиотека была проста в использовании, имела приятный и современный дизайн и позволяла в режиме реального времени обрабатывать введенные значения. Мы проанализировали каждую из библиотек по данным критериям и присовили баллы от 0 до 3 (где 0 - вообще не соответсвует критерию и 3 - полностью удовлетворяет критерию).

	Простота в использовании	Современный дизайн	Работа в режиме реального времени
Matplotlib	3	0	0
Dearpygui	1	3	3
Manim	2	2	2

Мы решили остановить свой выбор на библиотеке Dearpygui. Хоть нам и пришлось потратить изрядное количество сил и времени на то, что бы освоить ее функционал, однако она позволила нам создать интерфейс, содержащий в себе и кнопки с ползунками, и динамически масштабируемые графики.

3.2 Код для построения графиков зависимости скорости и высоты подъема от времени

```
1  import dearpygui.dearpygui as dpg
2  from math import exp
3
4  dpg.create_context()
5
6  ro_stat = 0.3
7  constant = 0.5
```

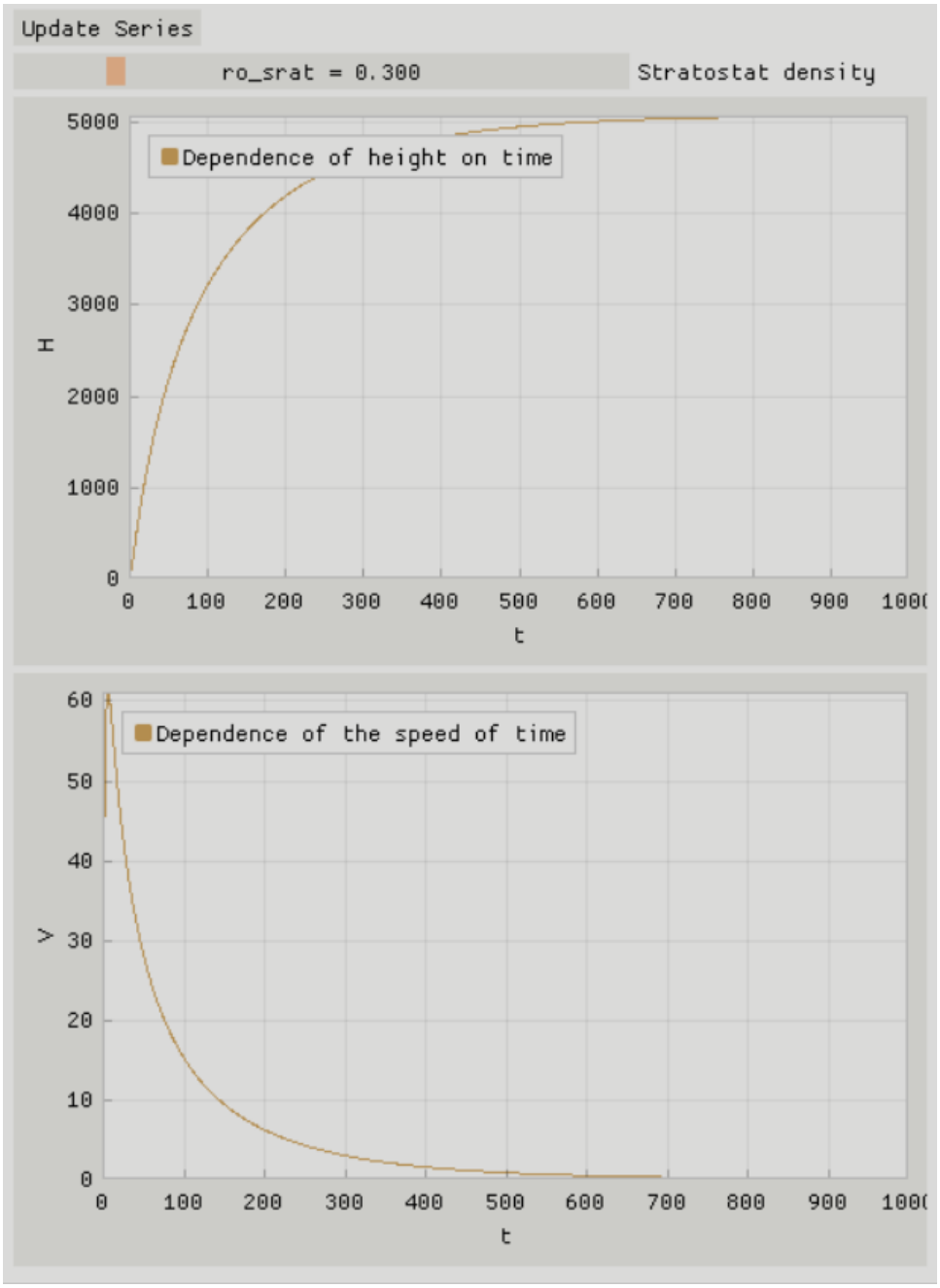
```
8   t = [0]
9   Y = [0]
10  V = [0]
11
12  def updateVandY(startV, startY):
13      dt = 0.5
14      ro_0 = 1.29
15      T_0 = 300
16      alpha = 0.0065
17      g = 9.81
18      R = 8.314
19      k = 0.47
20      myExp = exp(-dt*k)
21      ro_air = ro_0 * 10 ** (-0.125 * (startY/1000))
22      V = (g * (ro_air - ro_stat) * (1 - myExp) + k * ro_stat *
            myExp * startV) / (k * ro_stat)
23      Y = startY + V * dt
24      return V, Y
25
26  for i in range(2000):
27      newV, newY = updateVandY(V[i], Y[i])
28      V.append(newV)
29      Y.append(newY)
30      t.append((i + 1) * 0.5)
31
32  def log(sender, app_data, user_data):
33      print(f"sender: {sender}, \t app_data: {app_data}, \t
           user_data: {user_data}")
34      print(dpg.get_value(but))
35      global ro_stat
36      ro_stat = float(dpg.get_value(but))
37  def update_series():
38      t = [0]
39      Y = [0]
40      V = [0]
41      print("here")
42      for i in range(2000):
43          newV, newY = updateVandY(V[i], Y[i])
44          V.append(newV)
45          Y.append(newY)
46          t.append((i + 1) * 0.5)
47      print(ro_stat)
```

```
48          print(t)
49          print(Y)
50          dpg.set_value('series_tag', [t, Y])
51          dpg.set_value('vt_tag', [t, V])
52
53      with dpg.window(label="Tutorial", width=500, height=700):
54          dpg.add_button(label="Update Series", callback=
                update_series)
55          but = dpg.add_slider_float(label="Stratostat density",
                default_value=0.3 ,max_value=2.0, format="ro_srat =
                %.3f", callback=log)
56
57
58          def query(sender, app_data, user_data):
59              dpg.set_axis_limits("xaxis_tag2", app_data[0],
                    app_data[1])
60              dpg.set_axis_limits("yaxis_tag2", app_data[2],
                    app_data[3])
61
62
63          # plot 1
64          with dpg.plot(no_title=False, height=300, callback=query,
                query=True, no_menus=False, width=-1):
65              dpg.add_plot_legend()
66              dpg.add_plot_axis(dpg.mvXAxis, label="t")
67              dpg.add_plot_axis(dpg.mvYAxis, label="H")
68              dpg.add_line_series(t, Y, label="Dependence of height
                     on time", parent=dpg.last_item(), tag="series_tag
                    ")
69
70          # plot 2
71          with dpg.plot(no_title=True, height=-1, no_menus=False,
                width=-1):
72              dpg.add_plot_legend()
73              dpg.add_plot_axis(dpg.mvXAxis, label="t", tag="
                    xaxis_tag2")
74              dpg.add_plot_axis(dpg.mvYAxis, label="V", tag="
                    yaxis_tag2")
75              dpg.add_line_series(t, V, label="Dependence of the
                    speed of time",parent="yaxis_tag2", tag="vt_tag")
76
```

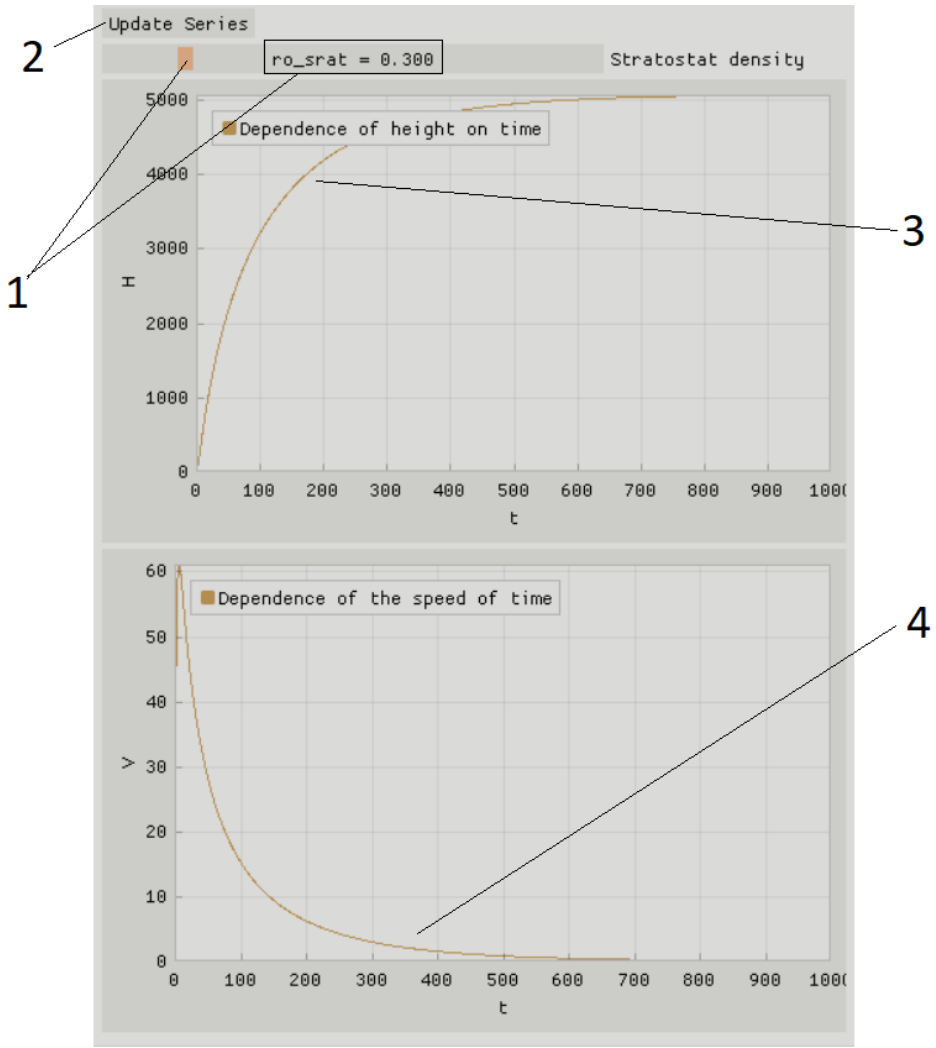
```
77  dpg.create_viewport(title='Custom Title', width=517, height
        =740)
78  dpg.setup_dearpygui()
79  dpg.show_viewport()
80  dpg.start_dearpygui()
81  dpg.destroy_context()
```

3.3 Результат работы программы

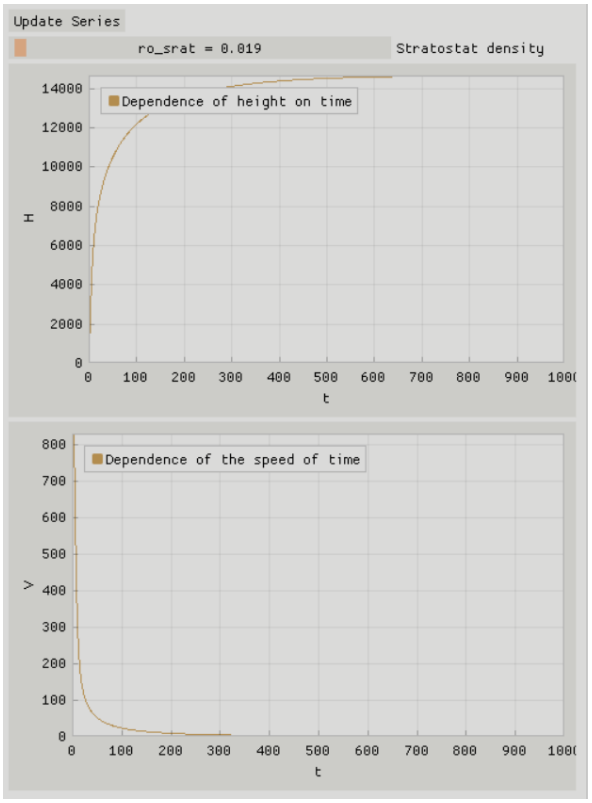
При запускке перед пользователем появляется окно с графиками, соответствующими значению плотности стратостата по умолчанию ($0.3\frac{kg}{m^3}$)

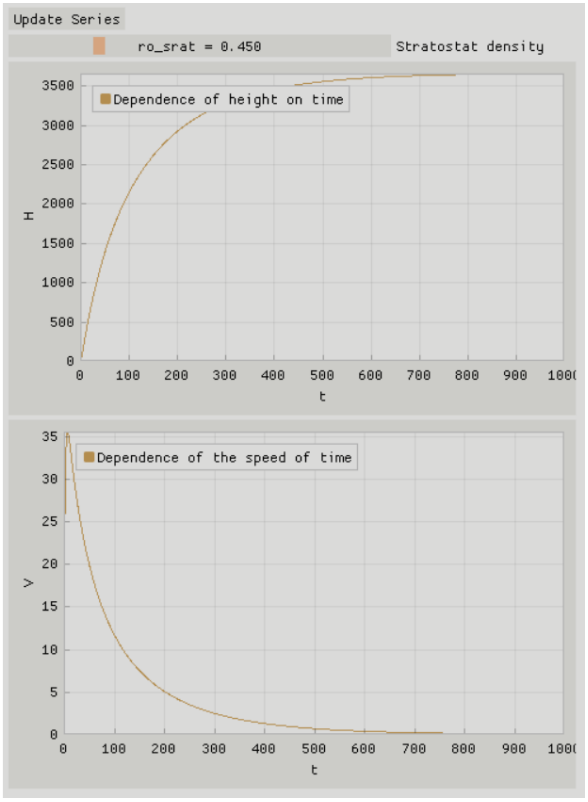


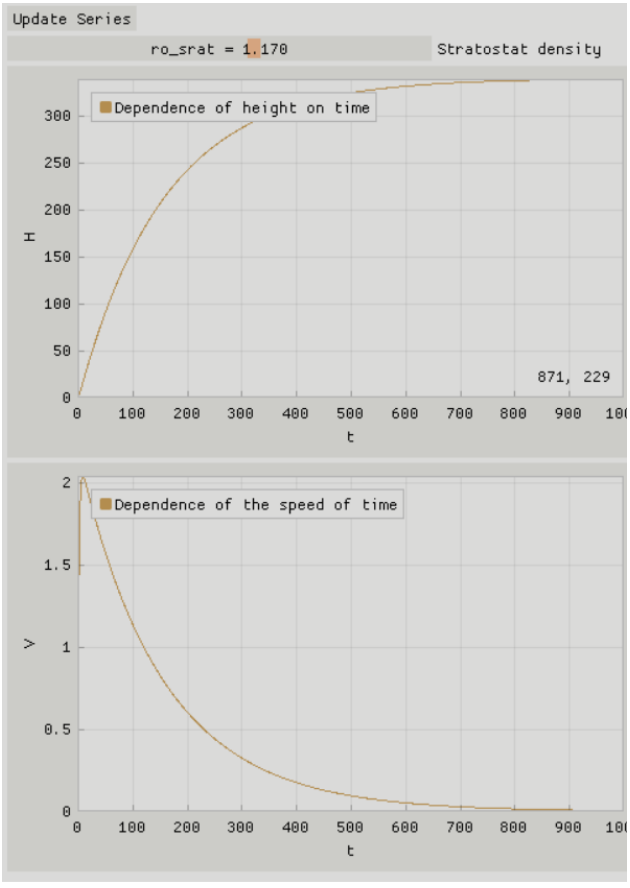
Далее необходимо выставить желаемое значение плотности стратостата при помощи ползунка(1) или двойным кликом ввести знаечние с клвиатуры. После этого пользователю необходимо нажать кнопку ”Update Series”(2), чтобф программа отредактировала графики. На экране появится два графика, верхний соотетствует изменению высоты стратостата с течением времени, нижний - скорости стратостату в данный момент времени. Стоит отметить, что графики являются не статичным изображением, а динмической фигурой. Благодаря этому пользователь может изменять масштаб осей, передвигать видимую область, а также получить точные координаты точки, наведя на нее курсор.



Мы запустили приложение и ввели несколько различных значений плтоности стратостата.


Update Series
ro_srat = 0.019
Stratostat density
Dependence of height on time
H
t
Dependence of the speed of time
V
t


Update Series
ro_srat = 0.450
Stratostat density
Dependence of height on time
H
t
Dependence of the speed of time
V
t


Update Series
ro_srat = 1.170
Stratostat density
Dependence of height on time
871, 229
H
t
Dependence of the speed of time
V
t

4 ВЫВОД

Подводя итоги проделанный работе, хочется сказать, что мы приобрели множество новых накыков и отточили старые. Например, мы уже умели решать дифференциальные уравнения, но проект позволил нам еще раз попрактиковаться в этом. Также нам пришлось изучить новую библеотеку и приемы программирования на языке Python, что сильно расширило наши навыки.
Самой полезной частью проекта, на наш взгляд была попытка соединить теоретические знания из разных научных областей в один готовый продукт, перенести физически существующие явления в виртуальный мир.
Стоит отметить, что наш проект масштабируем. В будущем мы можем повысить верхнюю границу высоты, до которой наше приложение может моделировать полет. Также мы можем добавить новые параметры стратостата, такие как масса его кабины и газвого баллона.