

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**Московский авиационный институт**

**(национальный исследовательский университет)**

Институт № 8 «Компьютерные науки и прикладная математика»

Кафедра 806 «Вычислительная математика и программирование»

**ОТЧЁТ**

По дисциплине:«Введение в авиационную и ракетно-космическую технику»

На тему: «Моделирование полёта «Спутник - 1»»

Оценка: Выполнили:

Подпись преподавателя: Группа М8О-101Б-24

\_\_\_\_\_\_\_\_ Птушка М. О.

\_\_\_\_\_\_\_\_ Коновалов Д.Д.

Москва, 2024

**ОГЛАВЛЕНИЕ**

[**ВВЕДЕНИЕ** 3](#_heading=h.30j0zll)

[**1.** Описание реальной миссии 5](#_heading=h.1fob9te)

[**1.1.** Краткая предыстория миссии запуска "Спутник-1" 5](#_heading=h.3znysh7)

[**1.2.** Миссия "Спутник - 1"](#_heading=h.nyvzyu5heu08) 6

[**1.3.** Разработка реального аппарата](#_heading=h.tyjcwt) 7

[**1.4.** Устройство реального аппарата](#_heading=h.3dy6vkm) 8

[**2.** Физико-математическая мо](#_heading=h.1t3h5sf)д[ель](#_heading=h.1t3h5sf) 10

[**2.1.** Системы координат 1](#_heading=h.luca0pavs3z3)0

[**2.2**. Силы и моменты, действующие на ракету в полете](#_heading=h.lexrwltx8lvr)14

**3.** Программная реализация 19

**4.** Моделирование в KSP 28

**ЗАКЛЮЧЕНИЕ** 31

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ** 32

**ПРИЛОЖЕНИЕ А** 33

**ПРИЛОЖЕНИЕ Б** 34

**ВВЕДЕНИЕ**

Название проекта: «Спутник-1».  
Название команды: «two men one ship».

Запуск первого искусственного спутника Земли «Спутник-1» является одним из важнейших этапов в истории освоения космоса. 4 октября 1957 года ракета-носитель «Спутник» вывела на орбиту первый искусственный спутник, положив начало космической эре. Проект команды «two men one ship» направлен на изучение и воспроизведение этой исторической миссии с использованием моделирования процесса запуска, полета и выхода на орбиту ракеты-носителя «Спутник».

Для реализации поставленных целей используется игра **Kerbal Space Program (KSP)**, которая предоставляет мощные инструменты для симуляции и проектирования космических аппаратов с учетом реальных физических законов. Эта платформа позволяет не только воссоздавать исторические события, но и проводить анализ технических аспектов космических миссий.

### **Актуальность проекта**

Проект имеет значительную научную и образовательную ценность. Во-первых, он дает возможность углубиться в изучение первых шагов человечества в космическом пространстве и понять, какие технологии и методы применялись на заре космонавтики. Во-вторых, современные симуляционные инструменты, такие как KSP, позволяют проверять точность расчетов и выявлять аспекты, важные для современных и будущих космических инициатив, где надежность играет ключевую роль.

### **Задачи проекта**

Для успешного выполнения проекта команда «two men one ship» формулирует следующие этапы:

1. Изучение исторических данных и технической документации, связанной с запуском «Спутник-1», а также характеристик ракеты-носителя.
2. Определение ролей и обязанностей участников команды для координации работы.
3. Разработка плана проекта с ключевыми этапами и сроками.
4. Создание математической модели, описывающей процесс запуска и выхода спутника на орбиту.
5. Проведение расчетов с использованием Python и специализированных библиотек.
6. Создание в KSP виртуальной модели ракеты-носителя «Спутник», учитывающей конструктивные особенности.
7. Симуляция миссии с полным воспроизведением этапов запуска, выхода на орбиту и работы спутника.
8. Сравнение результатов симуляции с данными математической модели для анализа их точности.
9. Подготовка итогового отчета с анализом результатов и выводами о точности модели и симуляции.

### **Таблица 1 – Состав команды «two men one ship»**

| **Участник** | **Роль** | **Задачи** |
| --- | --- | --- |
| Птушка М. О. | Физик/Математик | Формирование физической и математической модели |
| Коновалов Д.Д. | Лётчик/Программист | Оформление отчета, видео, программирование мат. модели, пилотирование ракеты |

Данный проект позволит объединить исторические данные, математическое моделирование и современные симуляционные инструменты для воспроизведения одной из важнейших миссий в истории освоения космоса.

**1. Описание реальной миссии**

**1.1. Краткая предыстория миссии запуска "Спутник-1"**

«Спу́тник - 1» — первый в мире искусственный спутник Земли, советский космический аппарат, запущенный на орбиту 4 октября 1957 года. Кодовое обозначение спутника — «ПС - 1»(«Простейший Спутник - 1»).Запуск был осуществлён с 5 - го научно - исследовательского полигона Министерства обороны СССР «Тюра - Там»(получившего впоследствии открытое наименование космодром «Байконур») на ракете - носителе «Спутник», созданной на базе межконтинентальной баллистической ракеты «Р - 7».

Полёту первого спутника предшествовала длительная работа многих учёных и конструкторов.Теорию реактивного движения одним из первых разработал в своих статьях Константин Эдуардович Циолковский.Проектированием спутника занимались Михаил Клавдиевич Тихонравов и его группа.Значительную роль в организации работ по созданию спутника и его запуска сыграл основоположник практической космонавтики Сергей Павлович Королёв.С ним работали такие учёные, как М.В.Келдыш, М.К.Тихонравов, М.С.Рязанский, О.Г.Ивановский, Н.С.Лидоренко, Г.Ю.Максимов, В.И.Лаппо, К.И.Грингауз, Б.С.Чекунов, А.В.Бухтияров, Н.А.Бересков и многие другие.

Дата запуска «Спутника - 1» является началом космической эры человечества, а в России ежегодно отмечается как памятный день Космических войск. В честь первого искусственного спутника Земли названа равнина на поверхности Плутона. [1]

**1.2. Миссия "Спутник - 1"**

Спутник ПС - 1 летал 92 дня, до 4 января 1958 года, совершив 1440 оборотов вокруг Земли (около 60 миллионов километров), а его радиопередатчики работали в течение двух недель после старта. [3]

**Цели запуска:**

•Проверка расчётов и основных технических решений, принятых для запуска.

•Ионосферные исследования прохождения радиоволн, излучаемых передатчиками спутника.

•Экспериментальное определение плотности верхних слоёв атмосферы по торможению спутника.

•Исследование условий работы аппаратуры.

Изучение характера радиосигнала и оптические наблюдения за орбитой позволили получить важные научные данные. Задача оптического наблюдения ИСЗ была поставлена коллективу Государственного астрономического института имени П.К.Штернберга МГУ В.Г.Куртом, П.В.Щегловым и В.Ф.Есиповым разработана методика наблюдений с точным определением координат спутника с временно́й привязкой.Для этой цели была приспособлена аэрофотосъёмочная камера НАФА с 10 - сантиметровым объективом, точные промежутки времени измерялись морским хронометром с электрическими контактами.После проявки плёнки треки спутника с помощью измерительного микроскопа «привязывались» к координатам звёзд, затем вручную(на механических счётных машинах) определяли шесть параметров орбиты. Время пересчёта занимало 30—60 минут. Фотографические наблюдения орбиты «Спутника - 1» проводились ежедневно, в течение двух недель В.Г.Куртом и П.В.Щегловым. Характер изменений орбиты позволил произвести предварительную оценку величины плотности атмосферы на орбитальных высотах, её высокое значение(порядка 108 атомов / см³) стало для геофизиков большой неожиданностью. Результаты измерения плотности высоких слоёв атмосферы позволили создать теорию торможения спутников, основы которой были заложены М.Л.Лидовым. [5]

**Параметры полёта:**

• Начало полёта — 4 октября 1957 года в 19:28:34 по Гринвичу

• Окончание полёта — 4 января 1958 года

• Масса аппарата — 83,6 кг

• Максимальный диаметр — 0,58 м

• Наклонение орбиты — 65,1°

• Период обращения — 96,2 минуты [6]

• Перигей — 228 км

• Апогей — 947 км

• Витков — 1440

**Технические Характеристики ракеты-носителя Р-7:**

• Максимальная дальность стрельбы — 8000

• Стартовая масса — 283 т

• Масса полезной нагрузки — 83,6 кг

• Масса топлива — 250 т

• Длина ракеты -31,4 м

• Диаметр ракеты — 1,2 м

• Тип головной части — моноблочная [7]

**1.3. Разработка реального аппарата**

Проектирование ПС - 1 началось в ноябре 1956 года, в начале сентября 1957 он прошёл окончательные испытания на вибростенде и в термокамере. Спутник был разработан как аппарат с двумя радиомаяками для проведения траекторных измерений. Диапазоны частот передатчиков простейшего спутника(20 МГц и 40 МГц) были выбраны так, чтобы сигнал спутника могли принимать радиолюбители без модернизации аппаратуры. [1]

Согласно воспоминаниям Г.М.Гречко, расчёты траектории вывода на орбиту «Спутника - 1» сначала проводились на электромеханических счётных машинах, по устройству аналогичных арифмометрам, и только для последних этапов расчётов применили ЭВМ БЭСМ - 1.

22 сентября в Тюра - Там прибыла ракета Р - 7 № 8К71ПС(изделие М1 - ПС «Союз»). По сравнению со штатными, она была значительно облегчена: массивная головная часть заменена переходом под спутник, снята аппаратура системы радиоуправления и одна из систем телеметрии, упрощена автоматика выключения двигателей; масса ракеты в результате была уменьшена на 7 тонн.

26 сентября Президиум ЦК КПСС постановил запуск спутника провести в середине октября.

2 октября Королёвым был подписан приказ о лётных испытаниях ПС - 1 и направлено в Москву уведомление о готовности. Ответных указаний не пришло, и Королёв самостоятельно принял решение о постановке ракеты со спутником на стартовую позицию. [4]

**1.4. Устройство реального аппарата**

Корпус первого спутника был изготовлен из алюминиево - магниевого сплава АМr - 6 толщиной 2 мм и состоял из двух полусфер диаметром 58.0 см, которые соединялись между собой 36 шпильками. Герметичность стыка обеспечивалась резиновой прокладкой.

Поверхность сферы «ПС - 1» была очень хорошо отполирована, это было сделано специально, для хорошего отражения лучей солнца и возможности визуального наблюдения спутника с земли.

С передней полусферы были расположены в перекрест угловые антенны вибраторы(1).Каждая антенна состояла из двух штырей, обращенных назад. Два штыря 2.4 метра(40, 002 МГц) и два штыря по 2.9 метра(20.002 МГц)

Плечи - штыри разводились пружинным механизмом после отделения от ракеты - носителя, в задней полусфере находился пяточный контакт(2), который после отделения спутника запускал автономную систему электропитания.

А вот, что находилось внутри этого серебристого шара.

Термореле(3), которое включало вентилятор(4), если температура повышалась выше + 30°С и отключало вентилятор, когда температура снижалась до примерно + 20°С. Соответственно имелся и воздуховод системы терморегулирования(5).Датчики температуры и давления(6).

Блок серебряно - цинковых аккумуляторов(7), а именно накальная батарея состояла из 5 элементов СЦД - 70 140 А / ч с напряжением 7.5 вольт, анодная батарея состояла из 86 элементов СПД - 18 30 А / ч с напряжением 160 вольт и имела отводы в 90 вольт и 10 вольт для питания экранных и пентодных сеток, так же отвод в 20 вольт для питания манипулятора. Вес всех электрохимических элементов питания составлял приблизительно 50 кг при общей массе спутника в 83, 6 кг. Ну и конечно же радиопередающее устройство(8), а именно бортовая радиостанция «ПРИБОР — Д200». Радиопередающее устройство излучало радиоволны на двух частотах : 20, 005 и 40, 002 МГц, поочерёдно(посылка сигнала одним передатчиком соответствовала паузе другого, переключение с периодичностью несколько десятых долей секунды осуществлялось электромеханическим реле).Для питания передатчиков, реле и вентилятора использовался набор серебряно - цинковых аккумуляторов(накальная батарея — 5 элементов СЦД - 70, 140 А·ч, 7, 5 В; анодная батарея — 86 элементов СЦД - 18, 30 А·ч, 130 В.

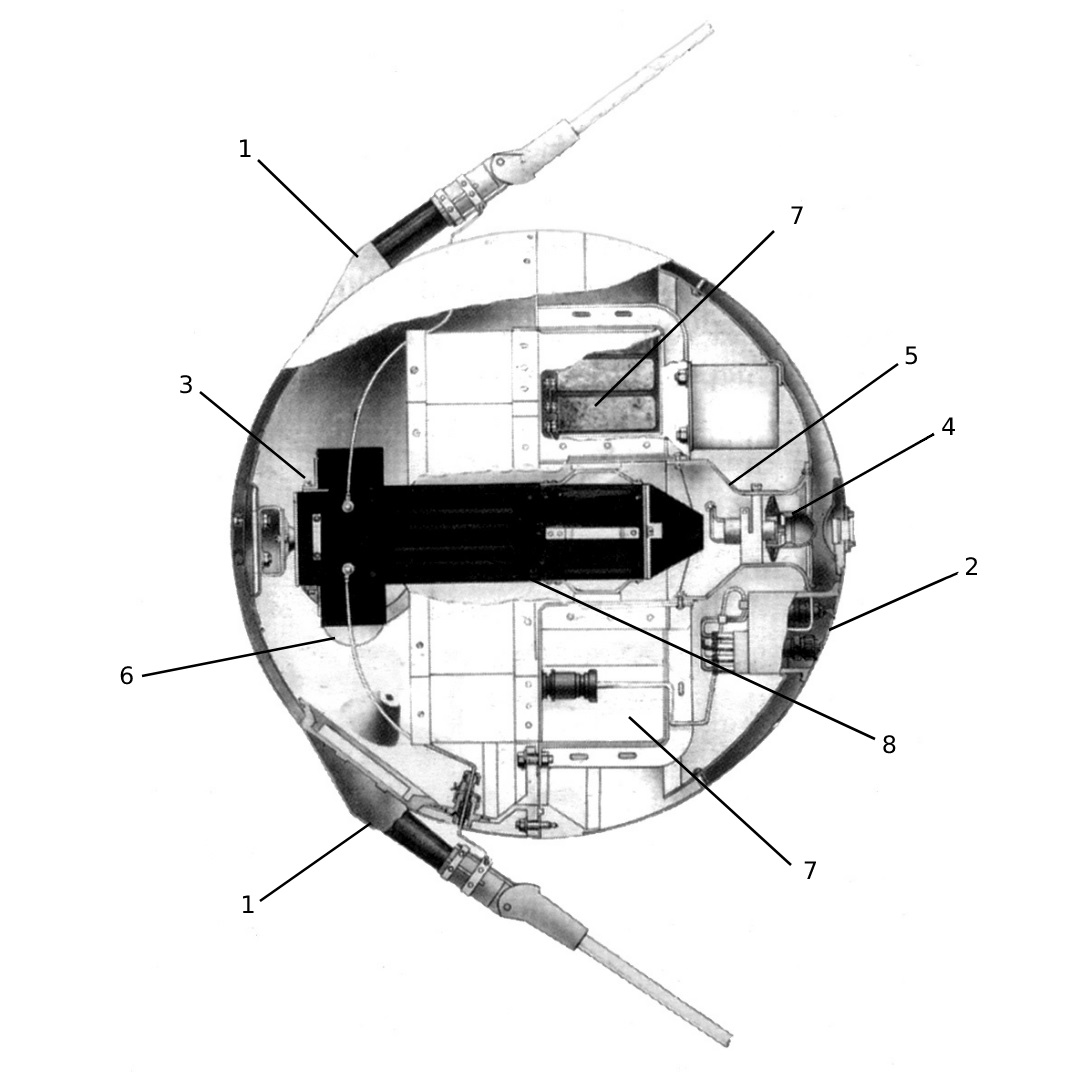


Рис.1 Сфера «ПС - 1» [2]

**2. Физико - Математическая модель**

**2.1 Расчёт массы спутника относительно времени**

1. **Общая масса ракеты**Масса ракеты состоит из трёх компонентов:

Где:

* — сухая масса конструкции ракеты (неизменная),
* — масса топлива, которая уменьшается со временем,
* — масса полезной нагрузки (спутника), неизменна.

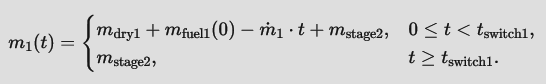
1. **Скорость расхода топлива**

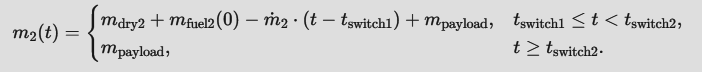
Масса топлива уменьшается со временем пропорционально скорости расхода топлива():

  
Для ступеней ракеты:  
  
Где:

* — начальная масса топлива на этапе,
* — скорость расхода топлива (кг/с).

1. **Масса ракеты на каждом этапе**

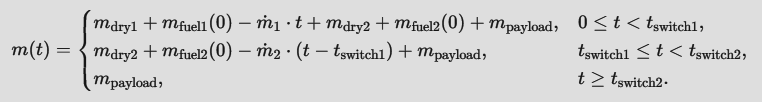
На первой ступени:  
  
На второй ступени:



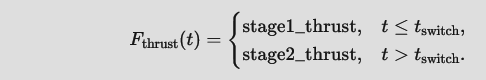
Здесь:

* — время срабатывания разделения ступеней,
* ​ — время полного сгорания топлива второй ступени.

1. **Функция зависимости массы ракеты от времени**

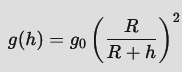


**2.2 Расчёт ускорения спутника относительно времени**

1. **Сила тяги:**Тяги ступени:  
   
2. **Сила сопротивления воздуха:**Зависит от текущей скорости v(t), плотности воздуха ρ(h), и площади поперечного сечения ракеты:  
   

Где:  
, а H = 8500 м – масштаб высоты.

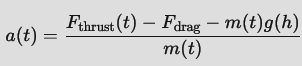
1. **Гравитация на высоте h:**

Гравитационное ускорение:  


Где R = 6371000 м – радиус Земли.

1. **Изменение массы ракеты:**Зависимость массы ракеты от времени было вычислено в п.2.1.
2. **Ускорение ракеты:**

По второму закону Ньютона:



**2.3 Расчёт скорости спутника относительно времени**

1. **Интегрируем ускорение:  
   **,

Где v(0) = 0 м/с.

**2.4 Скорость высоты спутника относительно времени**

1. **Интегрируем скорость:  
   **,  
   Где h(0) = 0 м.

**3. Программная реализация**

Данная программа предназначена для моделирования и анализа траектории ракеты с двумя ступенями. Она рассчитывает параметры движения ракеты (высота, скорость, угол наклона, масса) с учётом аэродинамического сопротивления, гравитации, и изменения массы ракеты во времени. Также генерируются и сохраняются графики, иллюстрирующие ключевые характеристики траектории.

1. Инициализация данных:
   * Определены физические константы (ускорение свободного падения, плотность воздуха, коэффициент сопротивления, площадь поперечного сечения ракеты).
   * Заданы параметры ступеней ракеты (масса, тяга, расход топлива).
   * Установлены начальные условия для переменных (координаты, скорость, масса, угол наклона тяги).
2. Основной цикл моделирования:
   * На каждом временном шаге пересчитываются гравитация, плотность воздуха, и параметры силы тяги.
   * Используются уравнения движения для вычисления ускорения, скорости, и положения ракеты.
   * Учтены отделение первой ступени и динамическое изменение площади поперечного сечения после её отделения.
   * Сохранение данных для построения графиков.
3. Сохранение графиков:
   * Все графики сохраняются в виде изображений (PNG).

import matplotlib

matplotlib.use('Agg') # Используем неблокирующий бэкенд

import numpy as np

import matplotlib.pyplot as plt

import math

def rocket\_trajectory():

# Константы

g0 = 9.81 # ускорение свободного падения на поверхности Земли, м/с^2

rho0 = 1.225 # плотность воздуха на уровне моря, кг/м^3

Cd = 0.742 # коэффициент сопротивления

A\_initial = 19.0 # начальная площадь поперечного сечения ракеты, м^2

R = 6371000 # радиус Земли, м

# Характеристики ступеней

stage1\_thrust\_per\_engine = 1000.3 \* 1000 # тяга одного двигателя первой ступени в невесомости, Н

stage1\_engines = 4 # количество двигателей первой ступени

stage1\_thrust = stage1\_engines \* stage1\_thrust\_per\_engine # общая тяга первой ступени, Н

stage1\_mass\_dry = 1740 \* stage1\_engines # сухая масса первой ступени, кг

stage1\_mass\_start = 44000 \* stage1\_engines # стартовая масса первой ступени, кг

stage2\_thrust = 941.4 \* 1000 # тяга второй ступени в невесомости, Н

stage2\_mass\_dry = 2840 # сухая масса второй ступени, кг

stage2\_mass\_start = 94000 # стартовая масса второй ступени, кг

# Расчёты

m0 = stage1\_mass\_start + stage2\_mass\_start # начальная масса ракеты, кг

fuel\_mass1 = stage1\_mass\_start - stage1\_mass\_dry # масса топлива первой ступени, кг

fuel\_mass2 = stage2\_mass\_start - stage2\_mass\_dry # масса топлива второй ступени, кг

burn\_rate1 = 335 \* stage1\_engines # скорость сгорания топлива первой ступени, кг/с

burn\_rate2 = 313.5 # скорость сгорания топлива второй ступени, кг/с

maneuver\_start\_altitude = 12000 # начало манёвра, м

maneuver\_end\_altitude = 110000 # конец манёвра, м

dt = 0.1 # шаг времени, с

t\_max = 400 # максимальное время, с

target\_periapsis = 200000 # целевая высота апоцентра, м

# Начальные условия

t = 0 # начальное время

x, y = 0, 0 # начальные координаты (м)

vx, vy = 0, 0 # начальные скорости (м/с)

angle = 90 # начальный угол наклона тяги (градусы)

mass = m0 # начальная масса

A = A\_initial # текущая площадь поперечного сечения

# Массивы для хранения данных

times = []

altitudeT = []

speedT = []

angleT = []

massT = []

def altitude():

return y

def vertical\_speed():

return vy

while t < t\_max and y >= 0:

# Пересчёт ускорения свободного падения и плотности воздуха

g = g0 \* (R / (R + y))\*\*2

rho = rho0 \* np.exp(-y / 8500) # аппроксимация снижения плотности воздуха

# Угол наклона тяги во время манёвра

if maneuver\_start\_altitude <= altitude() <= maneuver\_end\_altitude:

angle = math.asin(math.sqrt((maneuver\_end\_altitude - altitude()) / (maneuver\_end\_altitude - maneuver\_start\_altitude))) \* 57.2958

elif altitude() > maneuver\_end\_altitude:

# Управление ракетой после манёвра

if altitude() > target\_periapsis + 10000 and vertical\_speed() > 50:

print(-10, vertical\_speed(), altitude(), t)

angle = -10.0

else:

print(0, vertical\_speed(), altitude(), t)

angle = 0.0

angle\_rad = np.radians(angle)

# Силы и расход топлива

if fuel\_mass1 > 0:

thrust1 = stage1\_thrust

fuel\_mass1 -= burn\_rate1 \* dt

else:

thrust1 = 0

if fuel\_mass2 > 0:

thrust2 = stage2\_thrust

fuel\_mass2 -= burn\_rate2 \* dt

else:

thrust2 = 0

if fuel\_mass1 <= 0 and mass > stage2\_mass\_dry:

mass -= stage1\_mass\_dry # Отделение второй ступени

A /= 5 # Уменьшение площади поперечного сечения

thrust = thrust1 + thrust2

weight = mass \* g

# Расчёт скорости и предотвращение переполнения

v\_squared = vx\*\*2 + vy\*\*2

if v\_squared > 1e10: # Ограничение на скорость, чтобы избежать переполнения

v\_squared = 1e10

v = np.sqrt(v\_squared)

drag = 0.5 \* rho \* Cd \* A \* v\_squared

drag\_x = drag \* (vx / v) if v > 1e-6 else 0

drag\_y = drag \* (vy / v) if v > 1e-6 else 0

# Уравнения движения

thrust\_x = thrust \* np.cos(angle\_rad)

thrust\_y = thrust \* np.sin(angle\_rad)

ax = (thrust\_x - drag\_x) / mass

ay = (thrust\_y - drag\_y - weight) / mass

# Обновление скоростей и координат

vx += ax \* dt

vy += ay \* dt

x += vx \* dt

y += vy \* dt

# Обновление массы

mass = stage1\_mass\_dry + stage2\_mass\_dry + fuel\_mass1 + fuel\_mass2

# Обновление времени

t += dt

# Сохранение данных

times.append(t)

altitudeT.append(y)

speedT.append(v)

angleT.append(angle)

massT.append(mass)

# Сохранение графика в файл

plt.figure(figsize=(10, 6))

plt.plot(times, altitudeT, label="Высота", color='r')

plt.title("Теоретический график высоты от времени")

plt.xlim(0, 300) # Фиксированная шкала времени

plt.ylim(0, 350000) # Фиксированная шкала высоты

plt.xlabel("Время (с)")

plt.ylabel("Высота (м)")

plt.grid()

plt.legend()

plt.savefig('teoretical\_trajectory.png') # Сохранение графика в файл

plt.figure(figsize=(10, 6))

plt.plot(times, speedT, label="Скорость", color='r')

plt.title("Теоретический график скорости от времени")

plt.xlim(0, 300) # Фиксированная шкала времени

plt.ylim(0, 10000) # Фиксированная шкала высоты

plt.xlabel("Время (с)")

plt.ylabel("Скорость (м/c)")

plt.grid()

plt.legend()

plt.savefig('teoretical\_speed.png') # Сохранение графика в файл

plt.figure(figsize=(10, 6))

plt.plot(times, angleT, label="Угол", color='r')

plt.title("Теоретический график угла от времени")

plt.xlim(0, 300) # Фиксированная шкала времени

plt.ylim(-20, 100) # Фиксированная шкала высоты

plt.xlabel("Время (с)")

plt.ylabel("Угол (градусы)")

plt.grid()

plt.legend()

plt.savefig('teoretical\_angle.png') # Сохранение графика в файл

plt.figure(figsize=(10, 6))

plt.plot(times, massT, label="Масса", color='r')

plt.title("Теоретический график массы от времени")

plt.xlim(0, 300) # Фиксированная шкала времени

plt.ylim(0, 300000) # Фиксированная шкала высоты

plt.xlabel("Время (с)")

plt.ylabel("Масса (кг)")

plt.grid()

plt.legend()

plt.savefig('teoretical\_mass.png') # Сохранение графика в файл

if \_\_name\_\_ == "\_\_main\_\_":

rocket\_trajectory()

Далее следует программа автопилота,

import matplotlib

matplotlib.use('Agg') # Используем неблокирующий бэкенд

from time import sleep, time

import math

import krpc

import asyncio

import matplotlib.pyplot as plt

conn = krpc.connect(name='Launch into orbit')

vessel = conn.space\_center.active\_vessel

# Set up streams for telemetry

#ut = conn.add\_stream(getattr, conn.space\_center, 'ut')

altitude = conn.add\_stream(getattr, vessel.flight(), 'mean\_altitude', 'vertical\_speed')

vertical\_speed = conn.add\_stream(getattr, vessel.flight(vessel.orbit.body.reference\_frame), 'vertical\_speed')

apoapsis = conn.add\_stream(getattr, vessel.orbit, 'apoapsis\_altitude')

#stage\_1\_resources = vessel.resources\_in\_decouple\_stage(stage=1, cumulative=False)

#srb\_fuel = conn.add\_stream(stage\_1\_resources.amount, 'SolidFuel')

stage1\_engines = [engine for engine in vessel.parts.engines if 'stage1' in engine.part.tag]

#fuel\_tank = vessel.parts.with\_tag('accelerator')[0]

# Получение объекта ресурса LiquidFuel

#liquid\_fuel\_resource = fuel\_tank.resources.with\_resource('Kerosene')[0]

# Получение текущего количества топлива

#current\_fuel = liquid\_fuel\_resource.amount

#vessel.control.rcs = True

vessel.control.throttle = 0.0

target\_periapsis = 200000

maneuver\_start\_altitude = 12000

maneuver\_end\_altitude = 110000

target\_apoapsis = 900000

angle = 90.0

v\_orb = 7790

r\_earth = 6371 \* 10\*\*3

g = 9.81

orbit\_angle = 65.1

times = []

altitudeT = []

speedT = []

massT = []

angleT = []

ap = vessel.auto\_pilot

sleep(1)

print('Начало отсчёта')

sleep(1)

# Countdown...

print('3...')

sleep(1)

print('2...')

sleep(1)

print('1...')

sleep(1)

print('Запуск!')

start\_time = time()

flag\_accelerators\_in\_place = True

async def infinite\_printer():

while True:

global vessel, flag\_accelerators\_in\_place, angle

if flag\_accelerators\_in\_place == True and stage1\_engines[0].thrust == 0:

flag\_accelerators\_in\_place = False

vessel.control.activate\_next\_stage()

sleep(1)

ap.engage()

sleep(1)

ap.target\_pitch\_and\_heading(angle, 90.0 + orbit\_angle)

sleep(1)

print(stage1\_engines[0].thrust)

elapsed\_time = time() - start\_time

print(f"Время с запуска = {elapsed\_time:.2f}")

vessel = conn.space\_center.active\_vessel

current\_stage = vessel.control.current\_stage

print(f"Текущая активная ступень: {current\_stage}")

#print('altitude', altitude())

#print('apoapsis', vessel.orbit.apoapsis\_altitude)

#print('periapsis', vessel.orbit.periapsis\_altitude)

#print('speed', vessel.orbit.speed)

#print('Vspeed', vertical\_speed())

#print('VVspeed', vessel.surface\_velocity\_reference\_frame)

#print('VVVspeed', vessel.orbit.argument\_of\_periapsis)

#print('Orbitalspeed', vessel.orbit.orbital\_speed)

#print('Orbitalspeed100', vessel.orbit.orbital\_speed\_at(100))

#print('liquid\_fuel', vessel.resources.amount('LiquidFuel'))

#print('drag', vessel.flight().drag)

print('\n')

times.append(elapsed\_time)

altitudeT.append(altitude())

speedT.append(vessel.orbit.speed)

massT.append(vessel.mass)

angleT.append(angle)

await asyncio.sleep(0.5)

async def main():

global flag\_accelerators\_in\_place, angle

flag\_accelerators\_in\_place = True

asyncio.create\_task(infinite\_printer())

vessel.control.throttle = 0.1

vessel.control.activate\_next\_stage()

await asyncio.sleep(2.86)

vessel.control.activate\_next\_stage()

ap.engage()

ap.target\_pitch\_and\_heading(90.0, 90.0 + orbit\_angle)

print('Угол изменён на 90')

vessel.control.rcs = True

vessel.control.sas = True

while altitude() < maneuver\_start\_altitude:

await asyncio.sleep(0.1)

#ap.target\_pitch\_and\_heading(85.0, 90.0 + orbit\_angle)

#print('Угол изменён на 85')

angle = 90.0

acceleration = 10.0

t0 = time()

speed = 0

while angle > 0.0:

try:

angle = math.asin((math.sqrt(maneuver\_end\_altitude - altitude()) / math.sqrt(maneuver\_end\_altitude - maneuver\_start\_altitude))) \* 57.2958

except:

angle = 0.0

ap.target\_pitch\_and\_heading(angle, 90.0 + orbit\_angle)

print(f'Угол изменён на {angle}')

await asyncio.sleep(0.1)

ap.target\_pitch\_and\_heading(0.0, 90.0 + orbit\_angle)

while vessel.orbit.apoapsis\_altitude < target\_apoapsis or vessel.orbit.periapsis\_altitude < target\_periapsis:

if altitude() > target\_periapsis + 12000 and vertical\_speed() > 50:

angle = -10.0

ap.target\_pitch\_and\_heading(angle, 90.0 + orbit\_angle)

else:

angle = 0.0

ap.target\_pitch\_and\_heading(angle, 90.0 + orbit\_angle)

await asyncio.sleep(0.1)

vessel.control.throttle = 0.0

sleep(2)

vessel.control.activate\_next\_stage()

sleep(1)

vessel.control.activate\_next\_stage()

plt.figure(figsize=(10, 6))

plt.plot(times, altitudeT, label="Высота", color='b')

plt.title("Теоретический график высоты от времени")

plt.xlim(0, 300) # Фиксированная шкала времени

plt.ylim(0, 350000) # Фиксированная шкала высоты

plt.xlabel("Время (с)")

plt.ylabel("Высота (м)")

plt.grid()

plt.legend()

plt.savefig('real\_trajectory.png') # Сохранение графика в файл

plt.figure(figsize=(10, 6))

plt.plot(times, speedT, label="Скорость", color='b')

plt.title("Теоретический график скорости от времени")

plt.xlim(0, 300) # Фиксированная шкала времени

plt.ylim(0, 10000) # Фиксированная шкала высоты

plt.xlabel("Время (с)")

plt.ylabel("Скорость (м/c)")

plt.grid()

plt.legend()

plt.savefig('real\_speed.png') # Сохранение графика в файл

plt.figure(figsize=(10, 6))

plt.plot(times, angleT, label="Угол", color='b')

plt.title("Теоретический график угла от времени")

plt.xlim(0, 300) # Фиксированная шкала времени

plt.ylim(-20, 100) # Фиксированная шкала высоты

plt.xlabel("Время (с)")

plt.ylabel("Угол (градусы)")

plt.grid()

plt.legend()

plt.savefig('real\_angle.png') # Сохранение графика в файл

plt.figure(figsize=(10, 6))

plt.plot(times, massT, label="Масса", color='b')

plt.title("Теоретический график массы от времени")

plt.xlim(0, 300) # Фиксированная шкала времени

plt.ylim(0, 300000) # Фиксированная шкала высоты

plt.xlabel("Время (с)")

plt.ylabel("Масса (кг)")

plt.grid()

plt.legend()

plt.savefig('real\_mass.png') # Сохранение графика в файл

asyncio.run(main())

код предназначен для управления запуском ракеты с использованием интерфейса управления Kerbal Space Program (через библиотеку krpc). Программа осуществляет мониторинг телеметрии, управление ориентацией и тягой ракеты, а также строит графики основных параметров полёта, таких как высота, скорость, угол наклона и масса ракеты.

#### **Подключение к KRPC**

conn = krpc.connect(name='Launch into orbit')

vessel = conn.space\_center.active\_vessel

Программа устанавливает соединение с сервером KRPC и получает объект текущего активного космического аппарата (vessel).

#### **Настройка потоков данных**

altitude = conn.add\_stream(getattr, vessel.flight(), 'mean\_altitude', 'vertical\_speed')

vertical\_speed = conn.add\_stream(getattr, vessel.flight(vessel.orbit.body.reference\_frame), 'vertical\_speed')

apoapsis = conn.add\_stream(getattr, vessel.orbit, 'apoapsis\_altitude')

Потоки данных используются для получения телеметрии в режиме реального времени (высота, вертикальная скорость, апоцентр орбиты).

#### **Управление движением**

ap = vessel.auto\_pilot

ap.target\_pitch\_and\_heading(angle, 90.0 + orbit\_angle)

Автопилот отвечает за изменение угла наклона ракеты (pitch) и ориентации по курсу (heading).

#### **Логика последовательного выполнения**

Код реализует следующие этапы:

1. **Подготовка к запуску**:
   * Инициализация начальных параметров.
   * Печать обратного отсчёта.
2. **Старт**:
   * Управление запуском первой ступени.
   * Постепенное изменение угла наклона ракеты в зависимости от высоты и скорости.
3. **Выход на орбиту**:
   * Контроль апоапсиса и периаапсиса для достижения целевой орбиты.
   * Отключение двигателя после достижения требуемых параметров орбиты.
4. **Отделение ступеней**:
   * Управление последовательной активацией стадий ракеты.

#### **Сбор данных**

В ходе полёта записываются значения следующих параметров:

* Время (times)
* Высота (altitudeT)
* Скорость (speedT)
* Масса ракеты (massT)
* Угол наклона (angleT)

Асинхронность достигается использованием asyncio:

* **Основная задача**:

async def main()

Основной цикл управления полётом.

* **Вспомогательная задача**:

async def infinite\_printer()

Периодически выводит телеметрию в консоль и сохраняет данные.

Асинхронность позволяет обрабатывать управление и телеметрию одновременно, не блокируя основной поток.

Код строит графики изменения параметров от времени:

На выходе обе программы создают следующие графики:

1. Высота от времени: показывает изменение высоты ракеты на протяжении полёта.
2. Скорость от времени: отражает динамику разгона ракеты.
3. Угол наклона тяги от времени: иллюстрирует изменение направления тяги во время активного управления.
4. Масса ракеты от времени: демонстрирует расход топлива и изменения массы ракеты.

Синий график отображает реальные данные, а красный теоретические



Рисунок 1 – График угла от времени

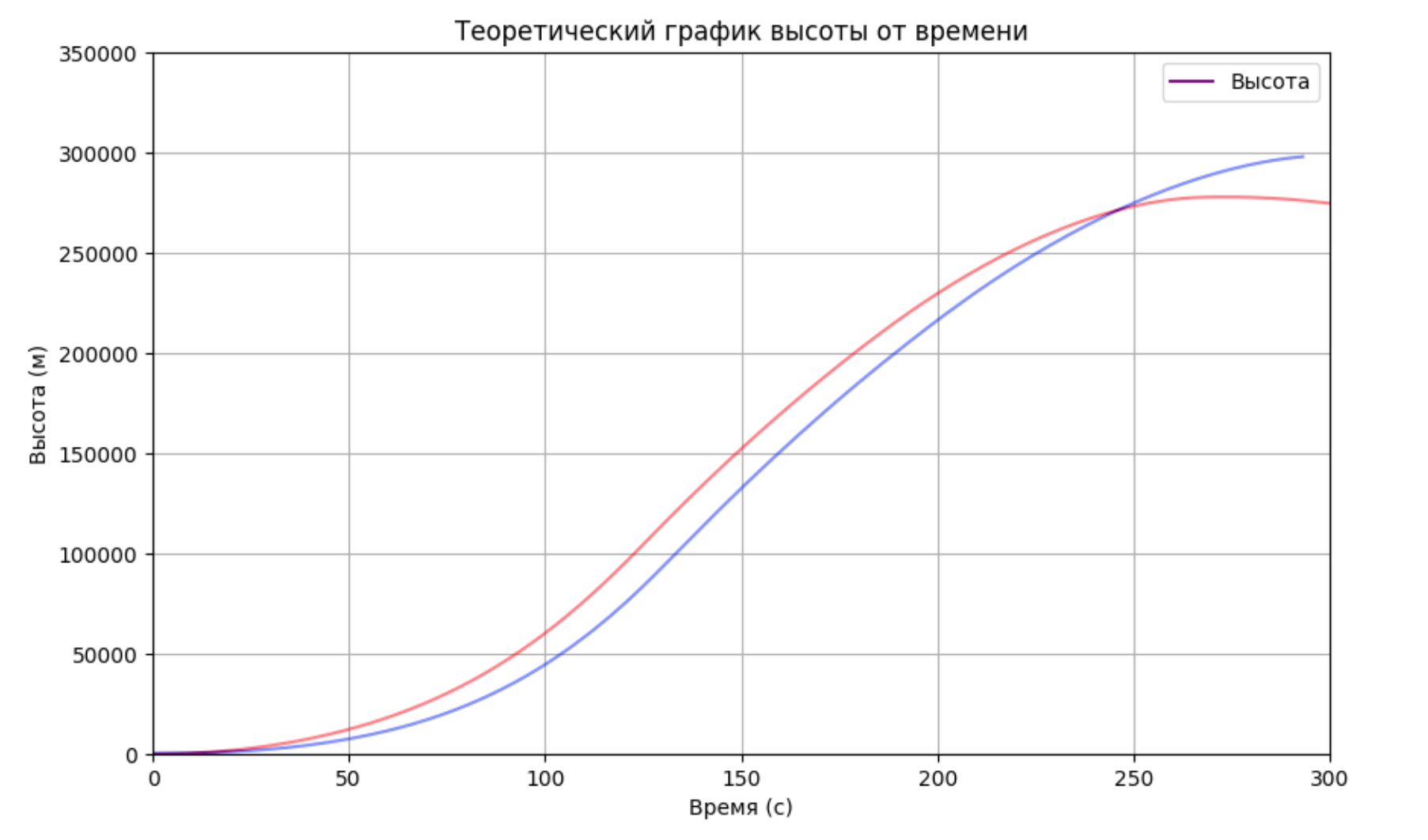


Рисунок 2 – График высоты от времени

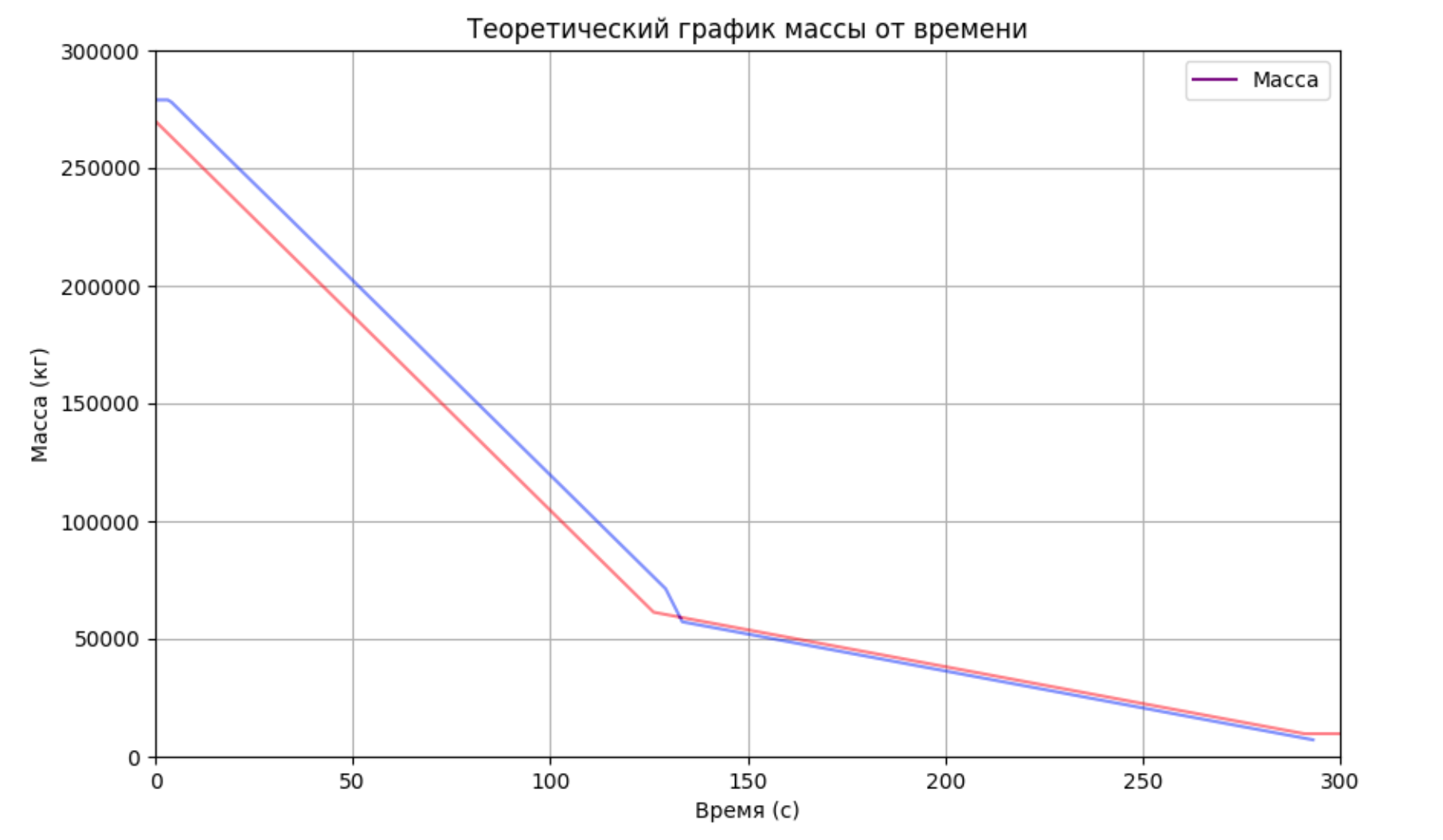


Рисунок 3 – График массы от времени

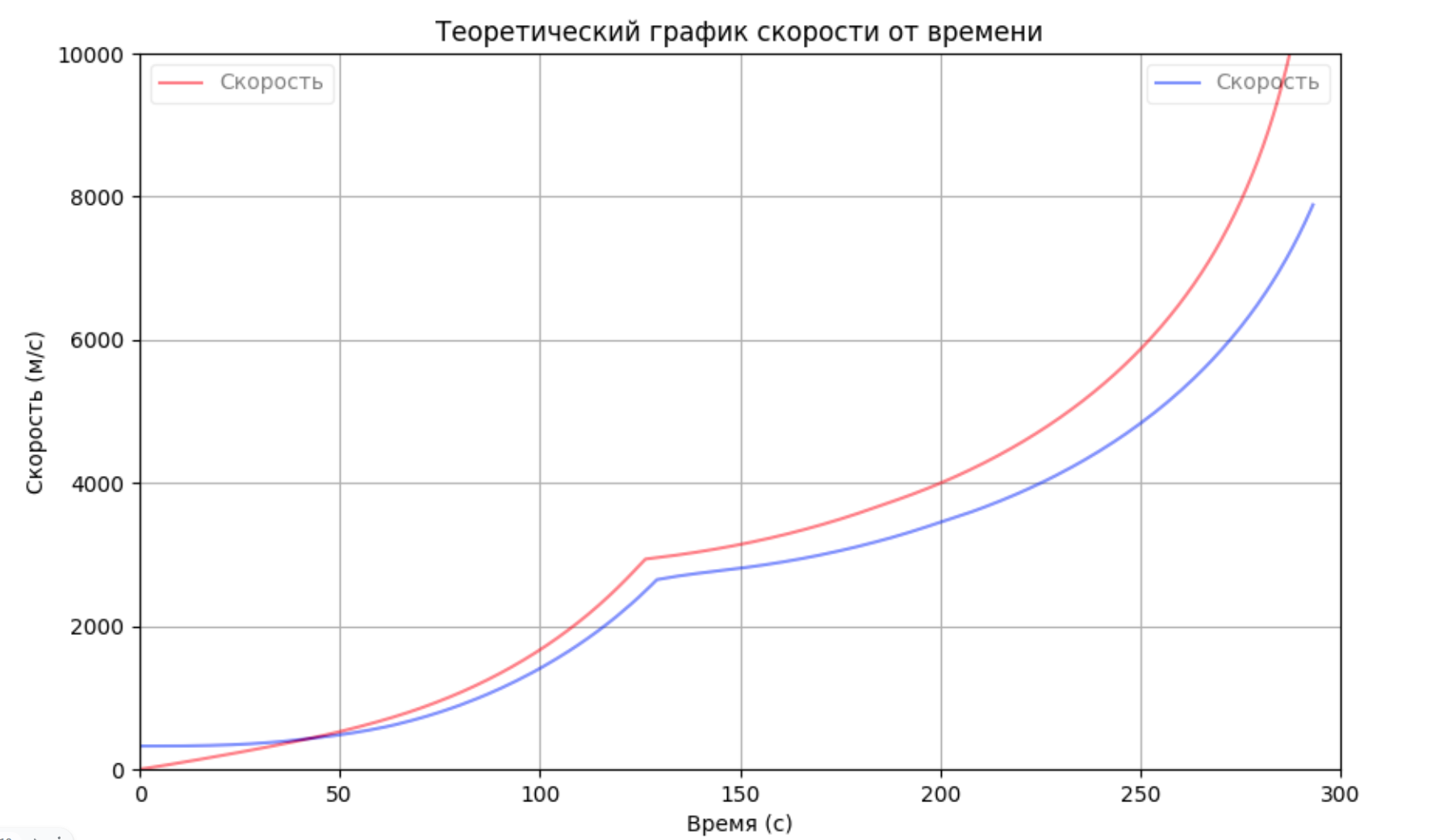


Рисунок 4 – График скорости от времени

**4. Моделирование в KSP**

Установим модификации Real Solar Planet (RSS) и RP-1, первая заменить внутриигровую систему на реальную солнечную, вторая же добавит реальные модели двигателей и позволит детальнее изменять форму корпуса ракеты.

Двигатель блока А - РД-108-8Д75 (Реальный РД-108-8Д75ПС)

Двигатели блоков Б, В, Г, Д - РД-107-8Д74 (Реальный РД-107-8Д74ПС)

Общая масса ракеты - 279,438 тонн (Реальная 267 тонн)

Масса блока А - 94 тонны (соответствует реальному)

Масса блоков Б, В, Г, Д (каждого) - 44 тонны (соответствует реальному)

Общая длина ракеты - 27,9 метров (реальная 29,167 метров)

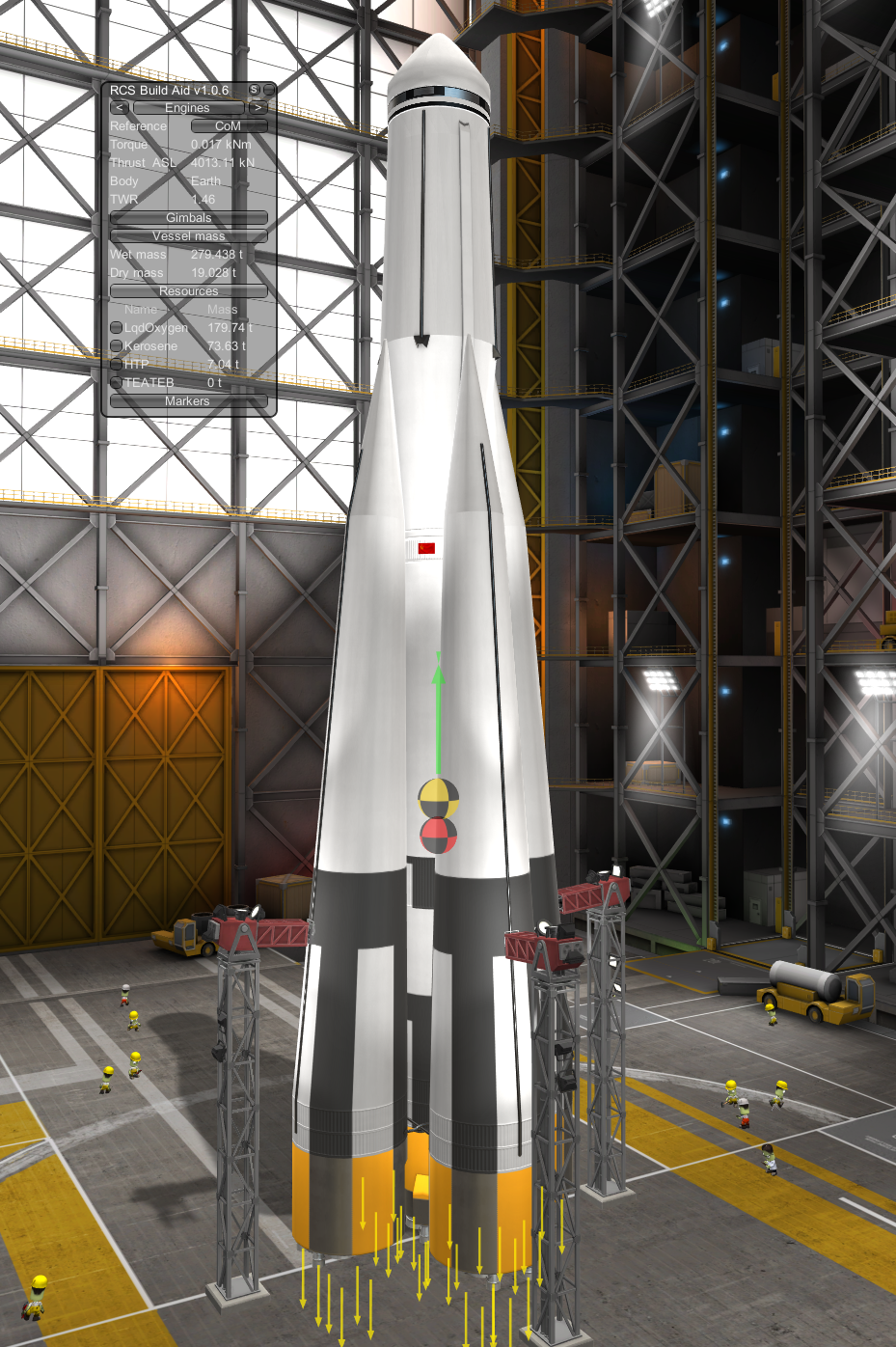


Рисунок 1 – Общий вид ракеты в ангаре



Рисунок 2 – Взлёт ракеты



Рисунок 3 – Ракета начинает маневрировать



Рисунок 4 – Ракета отделяет ступени



Рисунок 5 – Ракета достигает минимального перицентра

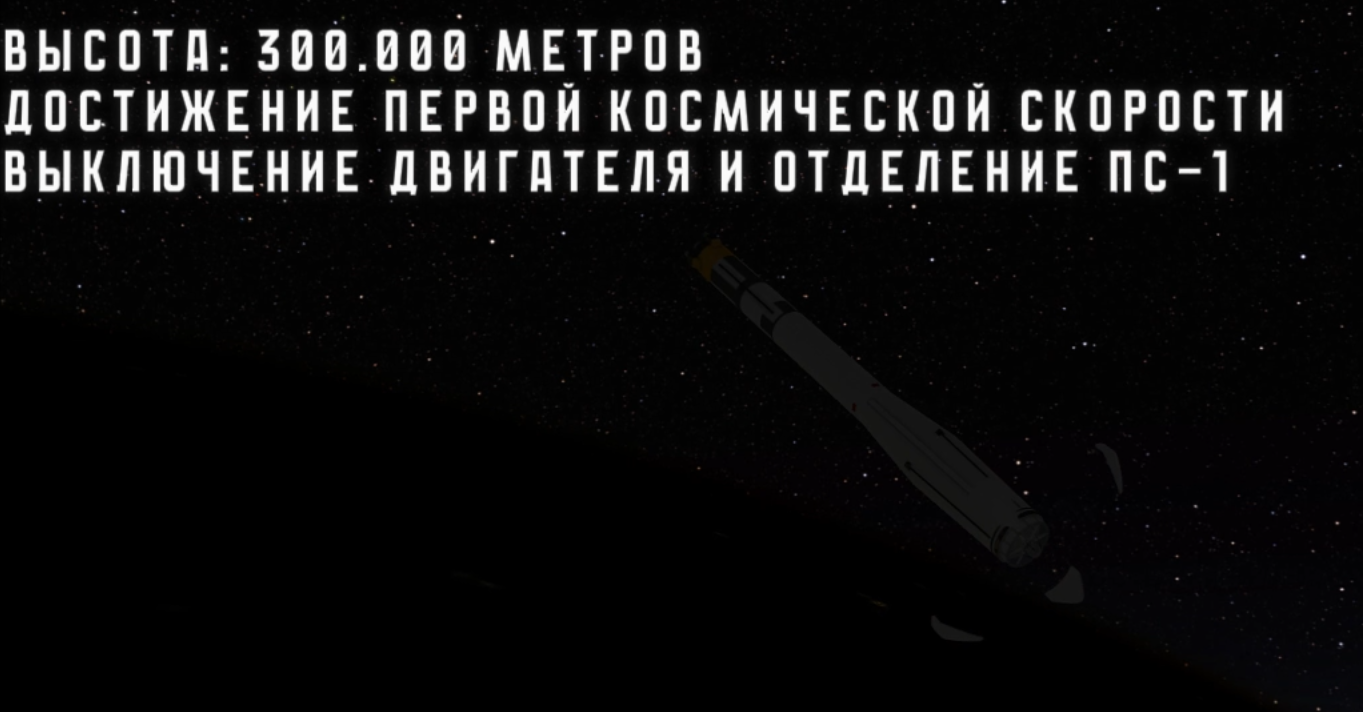


Рисунок 6 – Ракета заканчивает полёт и отделяет спутник

**ЗАКЛЮЧЕНИЕ**

В рамках выполнения проекта наша команда “two men one ship” успешно достигла поставленной цели — симулировать процесс вывода на орбиту первого искусственного спутника Земли «Спутник-1».

Для достижения этой цели мы провели всесторонний анализ данных реальной миссии, разработали физические формулы, создали математическую модель и на ее основе написали программный код для моделирования полета. Использование языка программирования Python позволило нам изучить **Matplotlib**, что обеспечило высокую точность и надежность расчетов.

Виртуальные модели ракеты-носителя и спутника были успешно воссозданы в симуляторе **Kerbal Space Program (KSP)**. Запуск ракеты-носителя, а также выполнение всех этапов миссии, прошли успешно. Однако мы зафиксировали незначительные отклонения от расчетных данных математической модели, которые составили около 5%. Эти отклонения, вероятно, вызваны несовершенством мат. модели и самой KSP

Анализ этих отклонений предоставил ценную информацию о возможных ошибках, которые стоит учитывать при проектировании будущих симуляций.

Проект предоставил нам возможность изучить и применить современные методы моделирования, а также решения прикладных задач, что сделало наш опыт более полным и полезным для дальнейшей работы.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1)PLUTO – Sputnik Planitia [Электронный ресурс] URL: https://planetarynames.wr.usgs.gov/Feature/15669

2)Спутник – 1 [Электронный ресурс] -  URL: https://r4uab.ru/istoriya-sputnikov/sputnik-1/ (Дата последнего обращения 30.11.24)

3)Первый в мире искусственный спутник Земли [Электронный ресурс] URL: https://www.roscosmos.ru/33800/ (Дата последнего обращения 30.12.24)

4)Степанов Б. Г. Передатчик первого ИСЗ // Радио. — 2013. — № 4. — С. 55—56

5)В. Г. Курт «Первые шаги отечественной астрономии из космоса», в сборнике «Первая космическая», стр. 239—241

6)Ю. А. Победоносцев. Триумф советской науки // Наука и жизнь. — 1957. - № 11. — С. 8-10.

7)Ракета-носитель Р7 [Электронный ресурс] URL : https://ruscosmos.narod.ru/KA/glavnaia/Rak\_nos/R7.htm

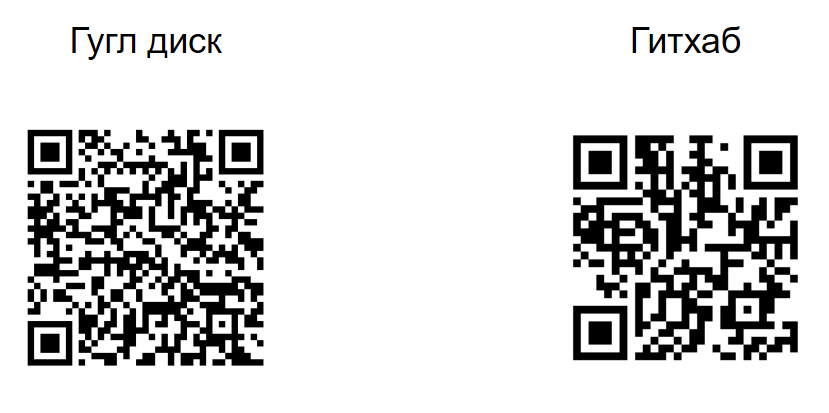
8)Kerbal Space Program Wiki // Kerbal Space Program. URL: https://wiki.kerbalspaceprogram.com/wiki/Main\_Page (дата обращения: 25.11.2024).

9)Физика (Яковлев И.В.) // mathus URL: https://mathus.ru/phys/book.pdf (дата обращения: 20.10.2024).

10)Matplotlib 3.9.3 documentation // Matplotlib URL: https://matplotlib.org/stable/index.html (дата обращения: 25.11.2024).

**Приложение A**

**Ссылки на онлайн ресурсы**

****