

Московский государственный технический университет им. Н.Э.

Баумана

Факультет: «Специальное машиностроение»

Кафедра СМ-3:

«Динамика и управление полетом ракет и космических аппаратов»

Работа на тему

«Влияние сопротивления верхней атмосферы на движение космического аппарата»

Выполнил студент группы СМ2-81_____Марков Д.М.

подпись дата

Проверил преподаватель: _____Голубев А.Г.

подпись дата

Москва, 2025

Оглавление

1. Задание	3
1.1 Формулировка задания	3
1.2 Исходные данные.....	3
2. Математическая модель.....	4
2.1 Допущения математической модели КА и его движения	4
2.2 Дифференциальные уравнения пространственного движение КА в прямоугольной системе координат	4
2.3 Допущения математической модели атмосферы	4
2.4 Система уравнений модели атмосферы	4
3. Результаты расчёта	6
4. Приложение	8
4.1. Код основной программы Python	8

1. Задание

1.1 Формулировка задания

1. Составить математическую модель возмущенного движения низкоорбитального космического аппарата под действием аэродинамического торможения верхней атмосферы.
2. Реализовать модель в программе на современном языке высокого уровня: C++, Python, Matlab или другом по согласованию с преподавателем.
3. Провести моделирование траектории численными методами.
4. Определить время существования КА на орбите выше 100 км. (Условие прекращения интегрирования – снижение высоты орбиты на 10 км от начальной.)

1.2 Исходные данные

В качестве исходной орбиты принять круговую околоземную орбиту высотой h по варианту. Гравитационный параметр Земли μ_z принять равным $\mu_z = 398600,45 \frac{\text{км}^3}{\text{с}^2}$, фигуру Земли считать сферической радиусом 6 371 км. Остальные параметры:

1. Высота $h = 276$ км;
2. Наклонение $i = 75^\circ$;
3. Индекс $F_{107} = 75$;
4. Баллистический коэффициент $\sigma = 0,004 \text{ м}^2/\text{кг}$.

2. Математическая модель

2.1 Допущения математической модели КА и его движения

1. Угловая ориентация КА не учитывается. Моделируется движение только центра масс.
2. Вращение Земли (и добавка к воздушной скорости КА, обусловленная вращением атмосферы вместе с Землёй) не учитывается;

2.2 Дифференциальные уравнения пространственного движение КА в прямоугольной системе координат

В математической модели используется АГСК. Следовательно, справедливы следующие уравнения, описывающие движение КА:

$$\frac{dx}{dt} = V_x; \quad \frac{dy}{dt} = V_y; \quad \frac{dz}{dt} = V_z; \quad (1)$$

$$\frac{dV_x}{dt} = -\mu_3 \frac{x}{r^3} + a_{\text{трмх}}; \quad \frac{dV_y}{dt} = -\mu_3 \frac{y}{r^3} + a_{\text{трму}}; \quad \frac{dV_z}{dt} = -\mu_3 \frac{z}{r^3} + a_{\text{трмz}}; \quad (2)$$

$$a_x = \frac{dV_x}{dt}; \quad a_y = \frac{dV_y}{dt}; \quad a_z = \frac{dV_z}{dt}. \quad (3)$$

Для систем (1)-(3): x, y, z – координаты КА, км; t – время, с; V_x, V_y, V_z – скорости КА, м/с; a_x, a_y, a_z – ускорения КА, м/с²; $r = \sqrt{x^2 + y^2 + z^2}$ – радиус-вектор КА, км; $a_{\text{трмх}}, a_{\text{трму}}, a_{\text{трмz}}$ – ускорения торможения атмосферы, действующее на КА, м/с².

Ускорение торможения атмосферы, обусловленные влиянием атмосферы на орбиту КА, направленные по осям АГСК, можно выразить следующим образом:

$$a_{\text{трмх}} = -\sigma \rho V^2 \frac{V_x}{V}; \quad a_{\text{трму}} = -\sigma \rho V^2 \frac{V_y}{V}; \quad a_{\text{трмz}} = -\sigma \rho V^2 \frac{V_z}{V} \quad (4)$$

Где $\sigma = \frac{C_x S_m}{2m}$ – баллистический коэффициент, $\rho(h, F_{107})$ – плотность, рассчитываемая по параметрам стандартной атмосферы; $V = \sqrt{V_x^2 + V_y^2 + V_z^2}$ – вектор скорости КА, м/с.

2.3 Допущения математической модели атмосферы

1. Слагаемые модели, зависящие от иных факторов, кроме индекса солнечной активности F_{107}/F_{81} и высоты, не учитываются (время года и суток, широта, индекс геомагнитной активности и пр.)
2. Осредненный исторический индекс принимается равным текущему

$$F_{107} = F_{81} = 75 \cdot 10^{-22} \frac{\text{Вт}}{\text{м}^2 \text{Гц}};$$

2.4 Система уравнений модели атмосферы

При расчете плотности атмосферы используют значения коэффициентов модели для уровня F_0 , ближайшего к значению F_{81} . Тогда $F_0 = 75 \cdot 10^{-22} \frac{\text{Вт}}{\text{м}^2 \text{Гц}}$.

Плотность модели атмосферы рассчитывается по ГОСТ Р 25645.166-2004, согласно которому при $h > 120$ км:

$$\rho = \rho_n K_0,$$

где $\rho_n = \rho_0 \exp(a_0 + a_1 h + a_2 h^2 + a_3 h^3 + a_4 h^4 + a_5 h^5 + a_6 h^6)$ – плотность ночной атмосферы;

$$K_0 = 1 + (l_0 + l_1 h + l_2 h^2 + l_3 h^3 + l_4 h^4) \frac{F_{81} - F_0}{F_0}$$

–изменение плотности атмосферы связанное с отклонением F_{81} от F_0 .

$$\text{Т.к. } F_0 = F_{81} = 75 \cdot 10^{-22} \frac{\text{Вт}}{\text{м}^2 \text{ Гц}}, \text{ то } K_0 = 1;$$

Здесь $\rho_0 = 1,64 \cdot 10^{-8} \text{ кг/м}^3$ – плотность ночной атмосферы на высоте 120 км;

Коэффициенты модели плотности атмосферы $a_0, a_1, a_2, a_3, a_4, a_5, a_6$, определяются при фиксированном уровне солнечной активности F_0 (см. таблица 1).

Таблица 1 – Коэффициенты модели плотности атмосферы при $F_0 = 75 \cdot 10^{-22} \frac{\text{Вт}}{\text{м}^2 \text{ Гц}}$.

Коэффициент		Значение
Обозначение	Размерность	
a_0	-	26,8629
a_1	км ⁻¹	-0,451674
a_2	км ⁻²	0,00290397
a_3	км ⁻³	– 1, 06953·10 ⁻⁵
a_4	км ⁻⁴	2, 21598·10 ⁻⁸
a_5	км ⁻⁵	– 2, 42941·10 ⁻¹¹
a_6	км ⁻⁶	1, 09926·10 ⁻¹⁴

3. Результаты расчёта

Полученные данные с шагом в 144000 секунд представлены в таблице 3.

Таблица 3 – Полученные результаты расчета

t, c	$x, км$	$y, км$	$z, км$	$V_x, км/с$	$V_y, км/с$	$V_z, км/с$	$h, км$
0 (t_0)	6647,000	0	0	0	2,0	7,5	276,000
144000	-1935,851	-1645,535	-6141,220	7,4	-0,6	-2,2	275,043
288000	-5649,042	905,694	3380,097	-4,1	-1,7	-6,4	274,074
432000	4654,040	1227,238	4580,115	-5,5	1,4	5,2	273,069
576000	3816,091	-1407,348	5252,296	6,3	1,2	4,3	272,025
720000	-5924,785	-776,999	-2899,799	3,5	-1,8	-6,7	270,960
864000	-2383,897	1604,222	5987,037	-7,2	-0,7	-2,7	269,866
1008000	6327,690	520,625	1943,000	-2,3	1,9	7,1	268,726
1152000	1912,892	-1645,306	-6140,366	7,4	0,6	2,2	267,545
1296000	-6295,766	-543,982	-2030,168	2,5	-1,9	-7,1	266,545
1334099 (t_k)	-4506,136	-1261,182	-4706,796	5,7	-1,4	-5,1	266,00

Таким образом, снижение высоты орбиты на 10 км от начальной произойдет за ≈ 15.5 дней. На основе полученных данных был построен график зависимости высоты КА над уровнем сферической Земли от времени $h(t)$, а также изображение траектории.

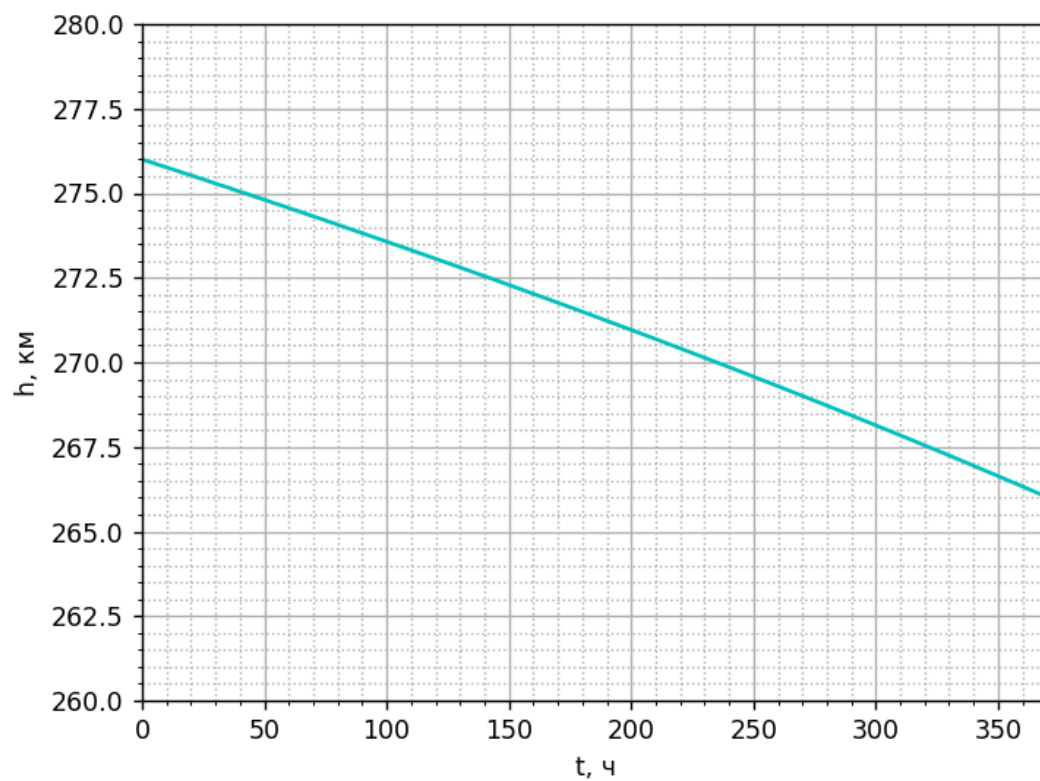


Рисунок 1 – График зависимости высоты КА над уровнем сферической Земли от времени.

Траектория КА

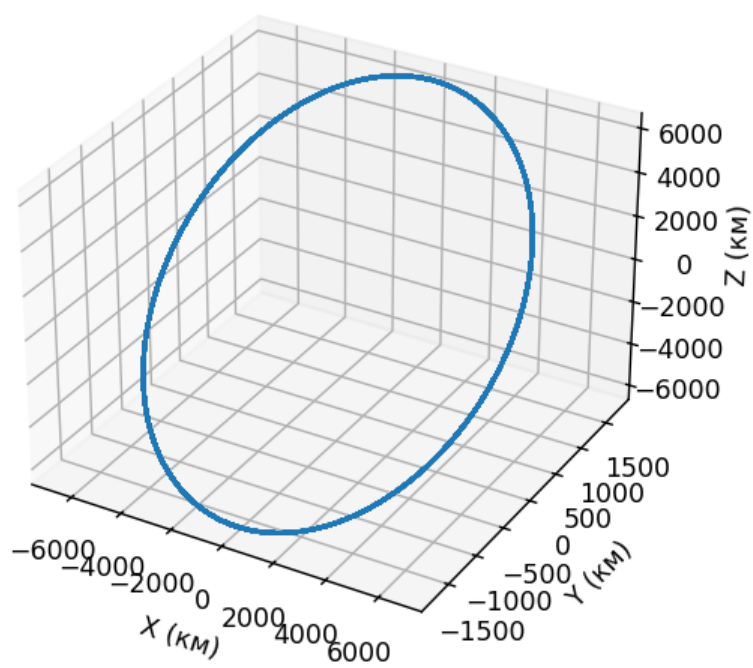


Рисунок 2 – Изображение траектории.

4. Приложение

4.1. Код основной программы Python

```
import numpy as np
import math
from matplotlib import pyplot as plt

MU_EARTH = 398600.45
R_EARTH = 6371

h_initial = 276
i_initial = np.radians(75)
F107 = 75
ballistic_coeff = 0.004

DATASET_GOST_ATMOSPHERE = {
    75: {
        'a0': 26.8629,
        'a1': -0.451674,
        'a2': 0.00290397,
        'a3': -1.06953e-5,
        'a4': 2.21598e-8,
        'a5': -2.42941e-11,
        'a6': 1.09926e-14
    }
}

def get_density(height_km, F107):
    K0 = 1
    rho_0 = 1.58868e-8
    a0 = DATASET_GOST_ATMOSPHERE[F107]['a0']
    a1 = DATASET_GOST_ATMOSPHERE[F107]['a1']
    a2 = DATASET_GOST_ATMOSPHERE[F107]['a2']
    a3 = DATASET_GOST_ATMOSPHERE[F107]['a3']
    a4 = DATASET_GOST_ATMOSPHERE[F107]['a4']
    a5 = DATASET_GOST_ATMOSPHERE[F107]['a5']
    a6 = DATASET_GOST_ATMOSPHERE[F107]['a6']
    rho_n = rho_0 * math.exp(a0 + a1 * height_km + a2 * height_km ** 2 + a3 *
height_km ** 3 + a4 * height_km ** 4 + a5 * height_km ** 5 + a6 * height_km **
6)
    density = rho_n * K0
    return density

def derivatives(statevec : np.array) -> np.array:
    derivs = np.zeros(shape=6)
    derivs[0:3] = statevec[3:6]
    r_cur = np.linalg.norm(statevec[0:3])
    h_cur = r_cur - R_EARTH
    v_cur = np.linalg.norm(statevec[3:6])
    derivs[3:6] = - MU_EARTH / r_cur ** 3 * statevec[0:3]
    rho = get_density(h_cur, F107)
    derivs[3:6] -= rho * ballistic_coeff * v_cur * statevec[3:6] * 1000
    return derivs

def runge_kutta4(statevec_i1 : np.array, step):
    statevec_i2 = np.zeros(shape=6)
    k1 = derivatives(statevec_i1)
    k2 = derivatives(statevec_i1 + step * k1 / 2)
```



```

k3 = derivatives(statevec_i1 + step * k2 / 2)
k4 = derivatives(statevec_i1 + step * k3)
statevec_i2 += statevec_i1 + step * (k1 + 2 * k2 + 2 * k3 + k4) / 6
return statevec_i2

def print_telemetry(time, statevec, height):
    print(
        f"t: {time:.0f} c; "
        f"X: {statevec[0]:.3f} км; "
        f"Y: {statevec[1]:.3f} км; "
        f"Z: {statevec[2]:.3f} км; "
        f"Vx: {statevec[3]:.1f} км/с; "
        f"Vy: {statevec[4]:.1f} км/с; "
        f"Vz: {statevec[5]:.1f} км/с; "
        f"Высота: {height:.3f} км"
    )

def solve(initial_statevec):
    current_state = initial_statevec.copy()
    time = 0.0
    revs = 0
    while True:
        h_current = np.linalg.norm(current_state[:3]) - R_EARTH
        if h_current <= target_height:
            break

        times.append(time)
        trajectory.append(current_state)
        heights.append(h_current)
        next_state = runge_kutta4(current_state, dt)
        if round(time, 2) % 21600 == 0:
            print_telemetry(time, current_state, h_current)

        if (current_state[2] <= 0 and next_state[2] >= 0):
            revs += 1

        time += dt
        current_state = next_state
        times.append(time)
        trajectory.append(current_state)
        heights.append(h_current)
        print_telemetry(time, current_state, h_current)
        print('РЕШЕНИЕ ЗАВЕРШЕНО')
        print(f'Число витков: {revs}')
        (print(f"Время снижения на 10 км: {times[-1] / 3600:.2f} часов"))

r_initial = R_EARTH + h_initial
v_initial = np.sqrt(MU_EARTH / r_initial)

initial_state = np.array([
    r_initial, 0, 0,
    0, v_initial * np.cos(i_initial), v_initial * np.sin(i_initial)
], dtype=np.float64)

dt = 1.0

times = []
trajectory = []
heights = []
target_height = h_initial - 10

```

```

solve(initial_state)

heights_plot = plt.figure().add_subplot()
plt.ticklabel_format(style='plain')
plt.xlabel("t, ч")
plt.ylabel("h, км")
plt.minorticks_on()
plt.xlim([0., 371.])
plt.ylim([260., 280.])
plt.grid(which = 'major')
plt.grid(which = 'minor', linestyle = ':')
heights_plot.plot([elem / 3600 for elem in times[:]], heights[:], color = 'c')

trajectory = np.array(trajectory, dtype=np.float64)
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.plot(trajectory[:, 0], trajectory[:, 1], trajectory[:, 2])
ax.set_xlabel('X (км)')
ax.set_ylabel('Y (км)')
ax.set_zlabel('Z (км)')
plt.title('Траектория КА')
ax.grid()
plt.show()

```

