

Отчет по лабораторной работе №3 по Мат Моделированию

1-2. Содержательная постановка задачи

Объект исследования:

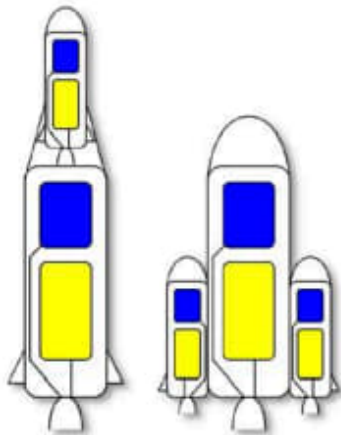
Объект исследования - многоступенчатая ракета (конкретно двух- и трехступенчатая).

Практическая задача - определить исследовать зависимость скорости двухступенчатой ракеты от соотношения масс топлива в каждой ступени; исследовать зависимость скорости движения трехступенчатой ракеты от времени для разных масс топлива в каждой ступени.

Модель должна давать количественный ответ - скорость многоступенчатой ракеты, в зависимости от соотношения масс топлива в каждой ступени; и скорость ракеты в зависимости от времени для разных масс топлива в каждой ступени.

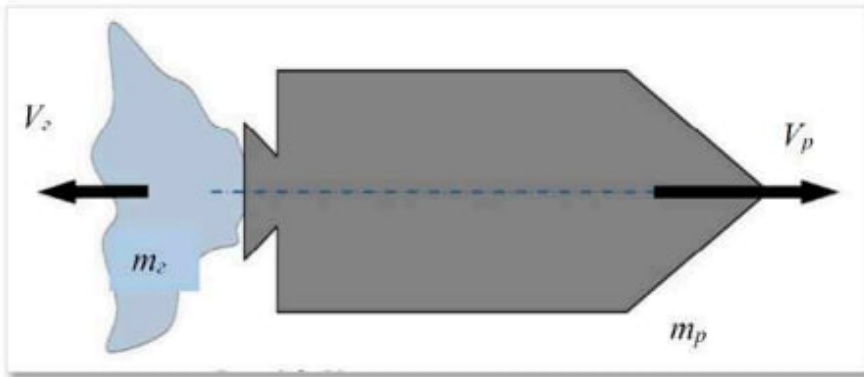
Исходные данные:

- m_p - масса ракеты;
- m_n - масса нагрузки;
- m_T - масса топлива в каждой ступени;
- V_T - скорость движения газов;
- V_0 - начальная скорость ракеты.



3. Концептуальная постановка задачи

Рассмотрим ракету в упрощенной форме, аналогично изображению на рисунке 3. При этом масса ракеты включает в себя два компонента: массу конструкции ракеты m_p и массу выбрасываемых газов m_g . В некоторый момент времени ракета движется с определенной скоростью V_p в то время как отработавшие газы выбрасываются со скоростью V_g . Предполагаем, что движение ракеты происходит в вакууме, то есть без учета сопротивления воздуха. Продукты сгорания покидают сопла ракеты в системе отсчета, связанной с неподвижной ракетой. При этом будем считать ракету материальной точкой.



4. Математическая постановка задачи

Рассмотрим движение двухступенчатой ракеты (для трехступенчатой аналогично).

Примем, что $V_0 = 0$. Для первой ступени $V_{p1} = V_0 + V_g * \ln(\frac{m_0}{m(t)}) = V_g * \ln(\frac{m_0}{m(t)})$, где $m_0 = m_p + m_{g1} + m_{g2} + m_n$. Пусть топливо в первой ступени израсходовано, тогда максимальная скорость для первой ступени будет $V_{p1_{max}} = V_g * \ln(\frac{m_0}{m_p + m_{g2} + m_n})$.

Для второй ступени $V_{p2} = V_{p1_{max}} + V_g * \ln(\frac{m_p + m_{g2} + m_n}{m_1(t)})$ и

$$V_{p2_{max}} = V_{p1_{max}} + V_g * \ln(\frac{m_p + m_{g2} + m_n}{m_p + m_n})$$

5. Реализация

```
```python
import numpy as np
import matplotlib.pyplot as plt

def rocket_two_stages(V_g, m_p, m_n, m_t, fuel_step=1000):
 """двухступенчатая ракета

 Args:
 V_g (float): скорость истекания газов.
 m_p (float): масса полезной нагрузки.
 m_n (float): масса несгоревшей конструкции (включая ступени).
 m_t (float): общая масса топлива.
 fuel_step (float): шаг уменьшения массы топлива (по умолчанию 1000).
 """
 m_total = m_p + m_n + m_t
 V0 = 0

 mt1 = 0.7 * m_t
 fuel_mass1, Vp1_list = [], []

 #первая ступень
 while mt1 > 0:
 Vp1 = V0 + V_g * np.log(m_total / (m_p + m_n + 0.3 * m_t + mt1))
 fuel_mass1.append(0.3 * m_t + mt1)
 Vp1_list.append(Vp1)
 mt1 -= fuel_step

 #вторая ступень
 mt2 = 0.3 * m_t
 fuel_mass2, Vp2_list = [], []
 while mt2 > 0:
 Vp2 = Vp1 + V_g * np.log((m_p + m_n + 0.3 * m_t) / (m_p + m_n + mt2))
 fuel_mass2.append(mt2)
 Vp2_list.append(Vp2)
 mt2 -= fuel_step

 plt.figure()
 plt.grid(True)
 plt.plot(fuel_mass1, Vp1_list, 'r*', label="Первая ступень")
 plt.plot(fuel_mass2, Vp2_list, 'b*', label="Вторая ступень")
 plt.xlabel('Масса оставшегося топлива, кг')
 plt.ylabel('Скорость ракеты, м/с')
 plt.title('Скорость двухступенчатой ракеты от массы топлива')
```

```
plt.legend()
plt.show()
```

```
def rocket_three_stages(V_g, m_p, m_n, m_t, fuel_step=1000):
 """трехступенчатая ракета

 args:
 V_g (float): скорость истекания газов.
 m_p (float): масса полезной нагрузки.
 m_n (float): масса несгоревшей конструкции (включая ступени).
 m_t (float): общая масса топлива.
 fuel_step (float): шаг уменьшения массы топлива (по умолчанию 1000).
 """
 m_total = m_p + m_n + m_t
 V0 = 0
 count = 0

 #первая ступень
 mt1 = 0.6 * m_t
 time1, Vp1_list = [], []
 while mt1 > 0:
 count += 1
 Vp1 = V0 + V_g * np.log(m_total / (m_p + m_n + 0.4 * m_t + mt1))
 time1.append(count)
 Vp1_list.append(Vp1)
 mt1 -= fuel_step

 #вторая ступень
 mt2 = 0.2 * m_t
 time2, Vp2_list = [], []
 while mt2 > 0:
 count += 1
 Vp2 = Vp1 + V_g * np.log((m_p + m_n + 0.4 * m_t) / (m_p + m_n + 0.2 * m_t + mt2))
 time2.append(count)
 Vp2_list.append(Vp2)
 mt2 -= fuel_step

 #третья ступень
 mt3 = 0.2 * m_t
 time3, Vp3_list = [], []
 while mt3 > 0:
 count += 1
 Vp3 = Vp2 + V_g * np.log((m_p + m_n + 0.2 * m_t) / (m_p + m_n + mt3))
 time3.append(count)
```

```
Vp3_list.append(Vp3)
mt3 -= fuel_step
```

```
plt.figure()
plt.grid(True)
plt.plot(time1, Vp1_list, 'r*', label="Первая ступень")
plt.plot(time2, Vp2_list, 'b*', label="Вторая ступень")
plt.plot(time3, Vp3_list, 'g*', label="Третья ступень")
plt.xlabel('Время, с')
plt.ylabel('Скорость ракеты, м/с')
plt.title('Скорость трехступенчатой ракеты от времени')
plt.legend()
plt.show()
```

## 6. Качественный анализ задачи

Выполним контроль размерности задач:

$$V_p = V_0 + V_r * \ln\left(\frac{m_0}{m(t)}\right) \Rightarrow \left[\frac{M}{C}\right] = \left[\frac{M}{C}\right] + \left[\frac{M}{C}\right] * \ln\left(\left[\frac{KG}{KG}\right]\right) = \left[\frac{M}{C}\right] + \left[\frac{M}{C}\right] = \left[\frac{M}{C}\right]$$

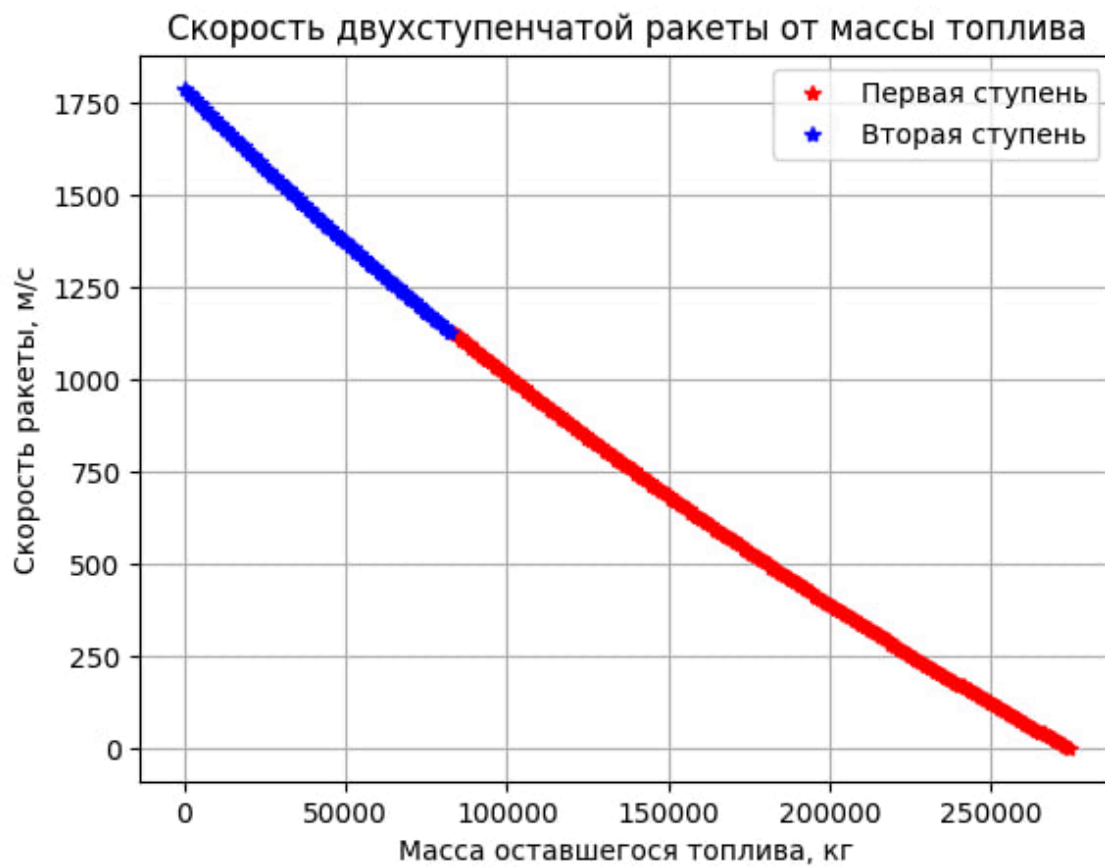
## 7. Численное исследование модели

При исследовании задачи были получены следующие решения

Для задачи двухступенчатой ракеты:

```
V_g = 3000
m_p = 300000
m_n = 34000
m_t = 274000
```

```
rocket_two_stages(V_g, m_p, m_n, m_t)
```



Для задачи трехступенчатой ракеты:

```
rocket_three_stages(V_g, m_p, m_n, m_t)
```

Скорость трехступенчатой ракеты от времени

