

## Содержание

<b>Введение .....</b>	<b>5</b>
<b>1.Постановка задачи .....</b>	<b>6</b>
<b>2.Решение краевой задачи .....</b>	<b>11</b>
<b>3.Решение задачи оптимизации.....</b>	<b>13</b>
<b>4.Определение потерь скорости СВ.....</b>	<b>16</b>
<b>5.Расчёт параметров промежуточных орбит .....</b>	<b>17</b>
<b>6.Исходные данные .....</b>	<b>18</b>
<b>7.Описание разработанного ПМО .....</b>	<b>19</b>
<b>8.Результаты решения задачи оптимизации .....</b>	<b>20</b>
<b>Выводы .....</b>	<b>37</b>
<b>Приложение А .....</b>	<b>39</b>

## **Введение**

Целью работы является освоение методов определения параметров программы управления средством выведения (СВ) при выведении полезного груза (ПГ) на целевую орбиту на примере решения задачи параметрической оптимизации программы управления вектором тяги двигательной установки (ДУ) при доставке ПГ с поверхности Луны на орбиту её искусственного спутника. Изучение влияния параметров программы управления вектором тяги на траекторию движения СВ.

Для достижения цели необходимо:

- разработать программно-математическое обеспечение, позволяющее определять программу управления вектором тяги ДУ СВ при компланарном выведении ПГ на орбиту искусственного спутника Луны (ИСЛ) с её поверхности, минимизирующую затраты компонентов топлива ДУ СВ;
- провести анализ влияния параметров программы управления вектором тяги ДУ на траекторию выведения ПГ и затраты компонентов топлива ДУ СВ.

## 1. Постановка задачи

Моделируется компланарное выведение ПГ с поверхности Луны на орбиту ИСЛ с заданными параметрами. Рассматривается схема выведения, содержащая пассивный участок. ДУ работает с момента старта  $t_0$  до некоторого момента времени  $t_1$ , после которого следует пассивный полёт средства выведения продолжительностью  $\Delta t_{\text{ПУТ}}$ . В момент времени  $t_2 = t_1 + \Delta t_{\text{ПУТ}}$  производится повторный запуск ДУ, которая работает далее вплоть до достижения граничных условий – до выхода на целевую орбиту. Закон управления ориентацией вектора тяги ДУ задан, определения требуют его параметры, а также параметры циклограммы работы ДУ:  $t_1$ ,  $\Delta t_{\text{ПУТ}}$ .

Выведение СВ на орбиту ИСЛ рассматривается в стартовой системе координат с учетом следующих допущений:

- атмосфера на Луне отсутствует (аэродинамические возмущения не учитываются);
- гравитационное поле Луны – центральное;
- вращение Луны не учитывается;
- возмущения от действия гравитационных полей Земли и других планет не учитываются;
- Луна – сфера радиуса  $R_m = 1738$  км;
- старт СВ осуществляется вертикально;
- система управления СВ считается идеальной, затраты массы компонентов топлива на программные развороты не учитываются;
- СВ рассматривается как материальная точка переменной массы.

Уравнения движения СВ в стартовой системе координат (Оху) имеют вид (1.1):

$$\left\{ \begin{array}{l} \frac{dV_x}{dt} = \frac{P}{m} \cos \vartheta + g_x, \\ \frac{dV_y}{dt} = \frac{P}{m} \sin \vartheta + g_y, \\ \frac{dx}{dt} = V_x, \\ \frac{dy}{dt} = V_y, \\ \frac{dm}{dt} = -\beta, \end{array} \right. \quad (1.1)$$

где  $m$  – масса СВ, кг;

$\beta$  – секундный расход топлива, кг/с;

$\vartheta$  – угол тангажа, рад;

$P = W_{\text{ист}}\beta$  – тяга ДУ, Н;

$W_{\text{ист}}$  – эффективная скорость истечения топлива,

$$g_x = \frac{\mu_m x}{[x^2 + (R_m + y)^2]^{\frac{3}{2}}},$$

$$g_y = \frac{\mu_m (R_m + y)}{[x^2 + (R_m + y)^2]^{\frac{3}{2}}} - \text{проекция ускорения от гравитационного поля Луны на}$$

соответствующие оси стартовой системы координат,  $\frac{\text{м}}{\text{с}^2}$ ;

$R_m = 1738$  км – радиус Луны;

$\mu_m = 4,903 \cdot 10^3 \cdot \text{км}^3/\text{с}^2$  – гравитационная постоянная Луны.

Ориентация вектора тяги ДУ в случае компланарного выведения определяется углом тангажа. Закон изменения угла тангажа на активных участках траектории при условии осуществления вертикального старта СВ имеет вид (1.2).

$$\left\{ \begin{array}{l} \frac{\pi}{2}, t \leq t_b; \\ \frac{\pi}{2} + \dot{\vartheta}_1(t - t_b), t_b < t \leq t_1; \\ \vartheta_2, t_2 < t \leq t_k, \end{array} \right. \quad (1.2)$$

где  $t_b$  – продолжительность вертикального участка, с;

$t_1$  – время начала пассивного участка траектории (выключения ДУ), с;

$t_2 = t_1 + \Delta t_{\text{ПУТ}}$  – время начала второго активного участка траектории (повторного включения ДУ), с;

$t_k$  – время достижения целевой орбиты, с;

$\dot{\vartheta}_1 = \frac{d\vartheta_1}{dt}$  – угловая скорость изменения угла тангажа на первом активном участке траектории, рад/с;

$\vartheta_2 = \text{const}$  – угол тангажа на втором активном участке траектории, рад.

Начальные условия на левом конце траектории (при  $t = t_0$ ):

$$x(t_0) = 0, y(t_0) = 0, V_x(t_0) = 0, V_y(t_0) = 0, m(t_0) = m_0.$$

Целевая орбита ИСЛ задается граничными условиями (1.3) на правом конце траектории выведения (при  $t = t_k$ ).

$$\begin{aligned} |\bar{r}(t_k)| &= R_{\text{ИСЛ}} = R_M + h_{\text{ИСЛ}}, \\ |\bar{V}(t_k)| &= V_{\text{ИСЛ}} = \sqrt{\frac{\mu_M}{R_{\text{ИСЛ}}}}, \\ (\bar{r}, \bar{V})(t_k) &= 0 \Leftrightarrow \theta(t_k) = 0, \end{aligned} \quad (1.3)$$

где  $r = \sqrt{x^2 + (R_M + y)^2}$  – радиус-вектор, проведенный из центра Луны;

$h_{\text{ИСЛ}}$  – высота круговой орбиты ИСЛ;

$\theta = \arcsin\left(\frac{(\bar{r}, \bar{V})}{|\bar{r}||\bar{V}|}\right)$  – угол наклона траектории к местному горизонту.

Продолжительность вертикального участка  $t_g$  считается заданной. Параметры закона управления вектором тяги  $\dot{\vartheta}_1, \vartheta_2, t_1, \Delta t_{\text{ПУТ}}$ , а также продолжительность полета  $t_k$  подлежат определению. Поскольку количество граничных условий на правом конце (1.3) меньше, чем количество параметров, подлежащих определению, то поставленная задача может иметь множество решений, среди которых требуется определить наилучшее по заданному критерию. Таким образом, можно сформулировать следующую задачу параметрической оптимизации:

Необходимо определить такие параметры закона управления ориентацией вектора тяги на АУТ и параметры циклограммы ДУ  $\dot{\vartheta}_1, \vartheta_2, t_1, \Delta t_{\text{ПУТ}}$ , а также продолжительность полета  $t_k$ , чтобы при заданных начальных условиях (1.2), на правом конце траектории выведения было обеспечено выполнение условий (1.3) при

минимальном возможном расходе компонентов топлива ДУ СВ. Т.е. выбор наилучшего решения производится исходя из критерия:

$$\Delta m_T = (m(t_0) - m(t_k)) \rightarrow \min. \quad (1.4)$$

Рабочий запас компонентов топлива ДУ СВ может варьироваться в некотором диапазоне, что, при учете заданной начальной массы СВ на поверхности Луны, означает возможность увеличения массы доставляемого на орбиту ИСЛ полезного груза за счет сокращения массы заправки компонентами топлива, потребными на осуществление выведения. Предельная возможная масса рабочего запаса топлива задана ( $\Delta m_{T \max}$ ).

Так как поставленная задача требует выполнения некоторого набора граничных условий на правом конце траектории, составной частью задачи параметрической оптимизации является краевая задача. При постановке краевой задачи количество граничных условий на правом конце траектории, как известно, должно соответствовать количеству искомых параметров управления. Так как время выведения  $t_k$  не задано, одно из условий (1.3) может быть выбрано для его определения – в качестве параметра останова численного интегрирования траектории движения. Таким образом, необходимо сформулировать краевую задачу размерности  $m = 2$ , где в качестве граничных условий выступают два оставшихся условия из (1.4). для этого среди неизвестных параметров закона управления ( $\dot{\vartheta}_1, \vartheta_2, t_1, \Delta t_{ПВТ}$ ) необходимо задать два параметра, которые будут обеспечивать выполнение этих краевых условий. Два оставшихся «свободных» параметра выбираются таким образом, чтобы обеспечить выполнение критерия (1.4).

Решение поставленной задачи должно быть получено с точностями не хуже:  $|r(t_k) - R_{ИСЛ}| < 10^{-3}$  м,  $|V(t_k) - V_{ИСЛ}| < 10^{-4}$  м/с,  $|\theta(t_k)| < 10^{-5}$  град,  $\varepsilon < 0.01$ . Точность для выбранного параметра останова численного интегрирования должна быть повышена относительно указанных.

Также необходимо с помощью разработанного ПМО **провести** численный расчет следующих производных:

$$\frac{\partial m_{\text{ПГ}}}{\partial m_{\text{конст}}}, \frac{\partial m_{\text{ПГ}}}{\partial W_{\text{ист}}},$$

где  $m_{\text{ПГ}} = m(t_k) - m_{\text{конст}}$  – масса полезного груза.

Далее для обеспечения оптимальной схемы выведения на целевую круговую орбиту необходимо найти параметры промежуточных орбит.

Для дополнительного анализа в работе приводятся величины потерь скоростей на управление и гравитационные потери скорости.

Интегрирование системы дифференциальных уравнений движения при решении задачи будет произведено методом Рунге-Кутты 4-го порядка с заданным шагом  $\Delta t$ . В качестве параметра останова численного интегрирования выбрана конечная скорость, поэтому для данного параметра повышаем точность:

$$|V(t_k) - V_{\text{ИСЛ}}| < 10^{-6} \text{ м/с.}$$

Для решения краевой задачи применяется модифицированный метод Ньютона с использованием формул численного дифференцирования.

В качестве параметров, обеспечивающих выполнение краевых условий:

$|r(t_k) - R_{\text{ИСЛ}}| < 10^{-3}$  м и  $|\theta(t_k)| < 10^{-5}$  град, были выбраны  $\dot{\vartheta}_1$  и  $\vartheta_2$ . В качестве «свободных» параметров управления остаются  $t_1, \Delta t_{\text{ПУТ}}$ .

## 2.Решение краевой задачи

Поставленную краевую задачу можно свести к решению системы нелинейных уравнений вида (2.1):

$$\Delta \bar{F}(\bar{U}) = 0, \quad (2.1)$$

где

$\bar{U} = [u_1 \ u_2 \ \dots \ u_m]^T$  – вектор-столбец неизвестных параметров краевой задачи;

$\Delta \bar{F}(\bar{U}) = [\Delta f_1 \ \Delta f_2 \ \dots \ \Delta f_m]^T$  – вектор-столбец невязок граничных условий.

Рассмотрим один из распространённых итерационных подходов решения нелинейных уравнений и их систем – **модифицированный метод Ньютона, применяющий формулы численного дифференцирования.**

Требуется решить систему нелинейных уравнений, включающую  $m$  неизвестных, следующего вида (2.2) :

$$\begin{cases} f_1(u_1, u_2, \dots, u_m) = 0, \\ f_2(u_1, u_2, \dots, u_m) = 0, \\ \dots \\ f_m(u_1, u_2, \dots, u_m) = 0. \end{cases} \quad (2.2)$$

В отличие от систем алгебраических линейных уравнений, использование прямых методов решения для нелинейных систем не представляется возможным. Во многих случаях затруднительно установить, существует ли решение системы (2.2) и каково их количество.

Обозначим решение системы (2.2)  $\bar{U}^* = [u_1^* \ u_2^* \ \dots \ u_m^*]^T$ .

Для его поиска необходимо применение итерационного метода, позволяющего на некоторой  $n$ -й итерации определить решение системы в заданной точностью  $\varepsilon$

$$: \|\bar{U}^n - \bar{U}^*\| < \varepsilon.$$

Разложение в ряд Тейлора в точке  $\bar{U}^i$ :

$$f_j(\bar{U}) \approx f_j(\bar{U}^i) + \sum_{q=1}^m \frac{\partial f_j(\bar{U}^i)}{\partial u_q} (u_q - u_q^i) \quad , j = 1, 2, \dots, m.$$

Вследствие этого система (2.2) на каждом итерационном шаге  $i$  преобразуется к системе линейных алгебраических уравнений (СЛАУ), которая в матричной форме записывается как (2.3):



$$\bar{F}(\bar{U}^i) + J(\bar{U}^i)(\bar{U} - \bar{U}^i) = 0, \quad (2.3)$$

где  $J = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \cdots & \frac{\partial f_1}{\partial u_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial u_1} & \cdots & \frac{\partial f_m}{\partial u_m} \end{bmatrix}$  – матрица Якоби.

Предположим, что матрица Якоби невырожденная, т.е. существует матрица  $J^{-1}$ . Тогда система (2.3) имеет единственное решение, которое принимается за очередное приближение  $\bar{U}^{i+1}$  к  $\bar{U}^*$ . Получаем расчётную итерационную формулу (2.4):

$$\bar{U}^{i+1} = \bar{U}^i + J^{-1}(\bar{U}^i) \left( -\bar{F}(\bar{U}^i) \right), \quad (2.4)$$

Производные в матрице Якоби можно заменить конечно-разностными отношениями.

В рассматриваемой задаче система уравнений для решения краевой задачи примет следующий вид:

$$\begin{cases} \Delta r(\dot{\vartheta}_1, \vartheta_2) = 0, \\ \Delta \theta(\dot{\vartheta}_1, \vartheta_2) = 0, \end{cases} \quad (2.5)$$

где  $\Delta r = r(t_k) - R_{\text{исл}}$  – невязки по радиус – вектору

$\Delta \theta = \theta(t_k) - \theta_{\text{исл}}$  – невязки по углу наклона вектора скорости к местному горизонту по окончании численного интегрирования.

Для решения матричного уравнения (2.4) используется следующая формула для нахождения обратной матрицы:

$$J^{-1} = \begin{bmatrix} \frac{\partial \Delta r}{\partial \dot{\vartheta}_1} & \frac{\partial \Delta r}{\partial \vartheta_2} \\ \frac{\partial \Delta \theta}{\partial \dot{\vartheta}_1} & \frac{\partial \Delta \theta}{\partial \vartheta_2} \end{bmatrix}^{-1} = \frac{1}{\det J} \begin{bmatrix} \frac{\partial \Delta \theta}{\partial \vartheta_2} & -\frac{\partial \Delta r}{\partial \vartheta_2} \\ -\frac{\partial \Delta \theta}{\partial \dot{\vartheta}_1} & \frac{\partial \Delta r}{\partial \dot{\vartheta}_1} \end{bmatrix}, \quad (2.6)$$

Где  $\det J = \frac{\partial \Delta r}{\partial \dot{\vartheta}_1} \frac{\partial \Delta \theta}{\partial \vartheta_2} - \frac{\partial \Delta r}{\partial \vartheta_2} \frac{\partial \Delta \theta}{\partial \dot{\vartheta}_1}$  – определитель матрицы Якоби.

Вычисление частных производных в матрице Якоби производилось по следующему алгоритму:

1. Интегрируются уравнения движения для заданных параметров  $\dot{\vartheta}_1$  и  $\vartheta_2$ , рассчитываются невязки  $\Delta r(\dot{\vartheta}_1, \vartheta_2), \Delta \theta(\dot{\vartheta}_1, \vartheta_2)$ .

2. Задаются малые приближения параметров  $\Delta\dot{\vartheta}_1$  и  $\Delta\vartheta_2$ . Затем проводится интегрирование для параметров  $\dot{\vartheta}_1 + \Delta\dot{\vartheta}_1, \vartheta_2$  и определяются невязки  $\Delta r(\dot{\vartheta}_1 + \Delta\dot{\vartheta}_1, \vartheta_2)$  и  $\Delta\theta(\dot{\vartheta}_1 + \Delta\dot{\vartheta}_1, \vartheta_2)$ . Аналогично для параметров  $\dot{\vartheta}_1$  и  $\Delta\vartheta_2 + \Delta\vartheta_2$  рассчитывают невязки  $\Delta r(\dot{\vartheta}_1, \vartheta_2 + \Delta\vartheta_2)$  и  $\Delta\theta(\dot{\vartheta}_1, \vartheta_2 + \Delta\vartheta_2)$ .

3. Значения частных производных аппроксимируются правыми разностями (2.7):

$$\begin{aligned}\frac{\partial \Delta r}{\partial \dot{\vartheta}_1} &\approx \frac{\Delta r(\dot{\vartheta}_1 + \Delta\dot{\vartheta}_1, \vartheta_2) - \Delta r(\dot{\vartheta}_1, \vartheta_2)}{\Delta\dot{\vartheta}_1}, \\ \frac{\partial \Delta r}{\partial \vartheta_2} &\approx \frac{\Delta r(\dot{\vartheta}_1, \vartheta_2 + \Delta\vartheta_2) - \Delta r(\dot{\vartheta}_1, \vartheta_2)}{\Delta\vartheta_2}, \\ \frac{\partial \Delta \theta}{\partial \dot{\vartheta}_1} &\approx \frac{\Delta \theta(\dot{\vartheta}_1 + \Delta\dot{\vartheta}_1, \vartheta_2) - \Delta \theta(\dot{\vartheta}_1, \vartheta_2)}{\Delta\dot{\vartheta}_1}, \\ \frac{\partial \Delta \theta}{\partial \vartheta_2} &\approx \frac{\Delta \theta(\dot{\vartheta}_1, \vartheta_2 + \Delta\vartheta_2) - \Delta \theta(\dot{\vartheta}_1, \vartheta_2)}{\Delta\vartheta_2},\end{aligned}\tag{2.7}$$

Следующее приближение параметров  $\dot{\vartheta}_1$  и  $\vartheta_2$  получается по следующей итерационной формуле:

$$\begin{bmatrix} \dot{\vartheta}_1 \\ \vartheta_2 \end{bmatrix}_{i+1} = \begin{bmatrix} \dot{\vartheta}_1 \\ \vartheta_2 \end{bmatrix}_i - \frac{1}{\det J} \begin{bmatrix} \frac{\partial \Delta \theta}{\partial \vartheta_2} & -\frac{\partial \Delta r}{\partial \vartheta_2} \\ -\frac{\partial \Delta \theta}{\partial \dot{\vartheta}_1} & \frac{\partial \Delta r}{\partial \dot{\vartheta}_1} \end{bmatrix}_i,\tag{2.8}$$

Расчёты повторяют до выполнения граничных условий:

$$|\Delta r| < 10^{-3} \text{ м и } |\Delta \theta| < 10^{-5} \text{ град.}$$

После анализа уравнений программы тангажа были приняты следующие начальные приближения:  $\dot{\vartheta}_1 = -0.003 \frac{\text{рад}}{\text{с}}$ ,  $\vartheta_2 = -0.3 \text{ рад}$ .

### 3.Решение задачи оптимизации

В рамках поставленной задачи требуется минимизировать расход топлива  $\Delta m_T = (m(t_0) - m(t_k)) \rightarrow \min$ . Для этого будет применяться метод градиентного спуска, так как он прост в реализации.

Роль управляющих параметров в данной оптимизационной постановке играют моменты начала  $t_1$  и завершения  $t_2$  полётного участка тяги (ПУТ).

Суть метода градиентного спуска с постоянным шагом заключается в том, что градиент  $\text{grad } f(\bar{X}^i)$  целевой функции  $f(\bar{X})$  в любой точке является вектором, указывающим направление наибольшего возрастания значения функции. Соответственно, антиградиент  $-\text{grad } f(\bar{X}^i)$  - направлен в сторону наибольшего убывания функции.

Тогда алгоритм поиска оптимальных параметров  $\bar{X}$  следующий:

1. Для заданного приближения параметров  $\bar{X}^i$  вычисляется градиент  $\text{grad } f(\bar{X}^i)$ .
2. Следующее приближение параметров рассчитывается по формуле (3.3):

$$\bar{X}^{i+1} = \bar{X}^i - \alpha \text{grad } f(\bar{X}^i) \quad (3.3)$$

где  $\alpha$  – коэффициент, задающий скорость сходимости метода (задаётся эмпирически).

3. Расчёты проводят до выполнения условия останова.

В соответствии с выбранным методом требуется определить вектор градиента. Поскольку явная аналитическая зависимость массы топлива от управляющих параметров отсутствует, градиент вычисляется численно.

Градиент – вектор, своим направлением указывающий направление наискорейшего роста функции.

Принимая, что масса топлива в данной задаче есть функция нескольких переменных, можно определить вектор градиента для неё с целью отыскания минимума – решения задачи оптимального выведения.

Расчёт массы топлива ведётся следующим образом:

1. Для текущего приближения управляющих параметров  $t_1$  и  $t_2$  находится решение граничной задачи, то есть определяются параметры  $\dot{\vartheta}_1, \vartheta_2$ , необходимые для попадания на целевую орбиту.
2. Используя параметры циклограммы, интегрируется система уравнений движения, в результате получаем значение массы для заданных управляющих параметров  $m(t_k)$  в конце выведения.
3. Тогда значение расхода топлива  $\Delta m_T(t_1, t_2) = (m(t_0) - m(t_k))$

Градиент  $\text{grad } \Delta m_T(t_1, t_2)$  рассчитывается по следующему алгоритму:

1. Определяется базовое значение массы топлива для заданных параметров.
2. Задаются малые приближения управляющих параметров  $\Delta t_1$  и  $\Delta t_2$ .  
Рассчитываются  $\Delta m_T(t_1 + \Delta t_1, t_2)$  и  $\Delta m_T(t_1, t_2 + \Delta t_2)$
3. Значения частных производных аппроксимируются правыми конечными разностями:

$$\begin{aligned}\frac{\partial m_T}{\partial t_1} &\approx \frac{\Delta m_T(t_1 + \Delta t_1, t_2) - \Delta m_T(t_1, t_2)}{\Delta t_1}, \\ \frac{\partial m_T}{\partial t_2} &\approx \frac{\Delta m_T(t_1, t_2 + \Delta t_2) - \Delta m_T(t_1, t_2)}{\Delta t_2}\end{aligned}\tag{3.1}$$

4. Определяем градиент:

$$\text{grad } \Delta m_T(t_1, t_2) = \begin{bmatrix} \frac{\partial m_T}{\partial t_1} \\ \frac{\partial m_T}{\partial t_2} \end{bmatrix}\tag{3.2}$$

Процесс оптимизации продолжается до удовлетворения критерия остановки:

$$|\text{grad } \Delta m_T(t_1, t_2)| < 0.0001.$$

Для решения задачи оптимизации эмпирически был подобран  $\alpha=50$ . Также были определены начальные приближения  $t_1=440\text{с}$ .  $t_2 = 850\text{с}$ .

## 4.Определение потерь скорости СВ

При выведении СВ на орбиту согласно заданной математической модели движения имеют место два вида потерь скорости: потери на управление и гравитационные потери. В данной работе они были найдены, при помощи метода трапеций, который был применён для интегралов (4.1-4.2)

$$\Delta V_{\text{упр}} = \int_0^{t_p} \frac{P}{m} (1 - \cos \alpha) dt \quad (4.1)$$

$$\Delta V_{\text{грав}} = \int_0^t g \cos \theta dt \quad (4.2)$$

Идеальную скорость можно найти по формуле Циолковского (4.3):

$$V_{\text{ц}} = V_0 + W_{\text{ист}} \ln \frac{m_0}{m_{\text{к}}}; \quad (4.3)$$

Итоговую скорость можно найти по формулам (4.4-4.6):

1-й  
АУТ:

$$V_{\text{к 1АУТ}} = W_{\text{ист}} \ln \frac{m_0}{m_{\text{к 1АУТ}}} - \int_0^{t_1} \frac{P}{m} (1 - \cos \alpha) dt - \int_0^{t_1} g \cos \theta dt, \quad (4.4)$$

ПУТ:

$$V_{\text{к ПУТ}} = V_{\text{к 1АУТ}} - \int_{t_1}^{t_2} g \cos \theta dt, \quad (4.5)$$

2-й  
АУТ:

$$V_{\text{к 1АУТ}} = V_{\text{к ПУТ}} + W_{\text{ист}} \ln \frac{m_{\text{к 1АУТ}}}{m_{\text{к 2АУТ}}} - \int_{t_2}^{t_{\text{к}}} \frac{P}{m} (1 - \cos \alpha) dt - \int_{t_2}^{t_{\text{к}}} g \cos \theta dt. \quad (4.6)$$

## 5. Расчёт параметров промежуточных орбит

После выключения двигательной установки в момент времени  $t_1$  СВ продолжит движение на ПУТ по промежуточной эллиптической орбите, параметры которой можно рассчитать по формулам (5.1).

$$\text{Модуль интеграла площадей: } c = (XV_{0y} - YV_{0x})^2;$$

$$\text{Параметр орбиты: } p = \frac{c^2}{\mu};$$

$$\text{Интеграл энергии: } h = \frac{1}{2} V_0^2 - \frac{\mu}{r_0};$$

$$\text{Большая полуось: } a = -\frac{\mu}{2h}; \quad (5.1)$$

$$\text{Эксцентриситет: } e = \sqrt{1 - \frac{p}{a}};$$

$$\text{Радиус перигея: } r_n = \frac{p}{1+e};$$

$$\text{Радиус апогея: } r_a = \frac{p}{1-e};$$

## 6. Исходные данные

$W_{ист} = 3370$  м/с эффективная скорость истечения топлива;

$m_0 = 3000$  кг начальная масса СВ;

$\Delta m_{Тmax} = 1525$  кг масса полезного груза;

$m_{конст} = 615$  кг масса конструкции СВ;

$t_e = 14$  с- продолжительность вертикального участка выведения;

$\Delta t = 0,1$ с - шаг интегрирования.

$P_1 = 10300$ Н;

$P_2 = 8160$ Н;

$h_{ИСЛ1} = 170$  км- высота 1-ой целевой орбиты ИСЛ;

$h_{ИСЛ2} = 310$  км- высота 2-ой целевой орбиты ИСЛ;

$h_{ИСЛ3} = 170$  км- высота 3-ой целевой орбиты ИСЛ;

$h_{ИСЛ4} = 200$  км- высота 4-ой целевой орбиты ИСЛ;

## 7.Описание разработанного ПМО

Программно – математическое обеспечение (ПМО) разработано в среде Visual Studio 2022 Community на языке программирования C++. Листинг ПМО представлен в приложении А.

В ПМО реализованы 4 основных функций:

-Функция `runge_kutt_4` осуществляет интегрирование заданной траектории методом Рунге-Кутты 4-го порядка используя 4 параметра закона управления  $\dot{\vartheta}_1, \vartheta_2, t_1, t_2$ . На вход функция получает вектор параметров управления и булеву переменную;

- Функция `metod` осуществляет вычисление параметров закона управления  $\dot{\vartheta}_1, \vartheta_2$  путем выполнения краевых условий (при фиксированных  $t_1$  и  $t_2$ ), позволяющих достичь краевых условий (1.3). Вычисления производятся по модифицированному методу Ньютона, применяющий формулы численного дифференцирования. На вход принимает также вектор параметров  $\dot{\vartheta}_1, \vartheta_2, t_1, t_2$  ;

- Функция `prime` осуществляет вычисление оптимальных параметров управления методом градиентного спуска с постоянным шагом;

- Функция `superprime` производит вычисление производной массы полезного груза по массе конструкции и производной массы полезного груза по скорости истечения;



## 8. Результаты решения задачи оптимизации

В таблицах 1-2 представлены результаты решения задачи оптимизации.

Таблица № 1 - Результаты решения задачи оптимизации для траекторий с  $P = 10300\text{Н}$

$P = 10.3 \text{ кН}$ $H = 150 \text{ км}$		$P = 10.3 \text{ кН}$ $H = 318 \text{ км}$	
$\dot{\vartheta}_1, \text{ рад/с}$	-0.0051	$\dot{\vartheta}_1, \text{ рад/с}$	-0.0044
$\vartheta_2, \text{ рад}$	-0.2487	$\vartheta_2, \text{ рад/с}$	-0.7126
$t_1, \text{ с}$	349.6232	$t_1, \text{ с}$	433.0594
$t_2, \text{ с}$	653.2985	$t_2, \text{ с}$	1342.6819
$m_k, \text{ кг}$	1588.386	$m_k, \text{ кг}$	1533.247
$m_{\text{топл}}, \text{ кг}$	1411.613	$m_{\text{топл}}, \text{ кг}$	1466,752

Таблица № 2 - Результаты решения задачи оптимизации для траекторий с  $P = 8160\text{Н}$

$P = 8.16 \text{ кН}$ $H = 150 \text{ км}$		$P = 8.16 \text{ кН}$ $H = 200 \text{ км}$	
$\dot{\vartheta}_1, \text{ рад/с}$	-0.0036	$\dot{\vartheta}_1, \text{ рад/с}$	-0.0035
$\vartheta_2, \text{ рад}$	-0.3385	$\vartheta_2, \text{ рад}$	-0.6151
$t_1, \text{ с}$	503.1250	$t_1, \text{ с}$	555.3703
$t_2, \text{ с}$	811.5474	$t_2, \text{ с}$	1175.4196
$m_k, \text{ кг}$	1566.793	$m_k, \text{ кг}$	1548.403
$m_{\text{топл}}, \text{ кг}$	1433.206	$m_{\text{топл}}, \text{ кг}$	1451.597

Анализируя полученные данные можно заметить, что при одинаковой тяге протяженность ПУТ больше для более высоких орбит.

При одинаковой высоте целевой орбиты, скорость изменения угла тангажа интенсивнее для орбиты с большей тягой.

Убедится в правильности решения задачи оптимизации можно, построив графики зависимости массы потраченного топлива в зависимости от  $t_1$  и  $t_2$ , включая оптимизированные параметры.

На рисунках 1-2 представлены данные графики.

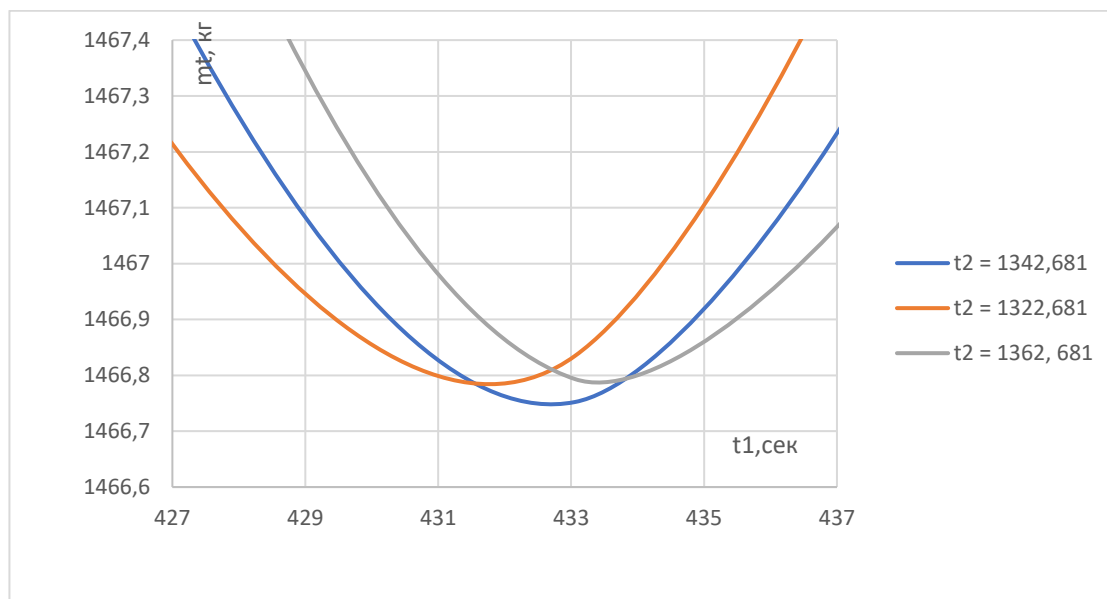


Рисунок 1 – Графики зависимости массы потраченного топлива от времени  $t_1$ , при неизменном  $t_2$

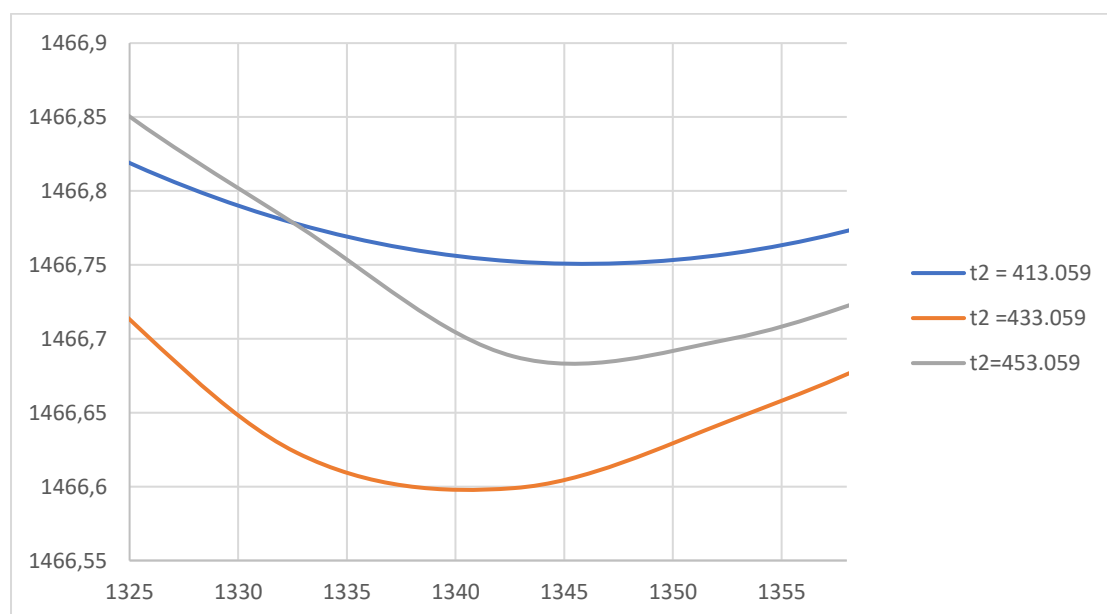


Рисунок 2 – Графики зависимости массы потраченного топлива от времени  $t_2$ , при неизменном  $t_1$

Проанализировав графики, можно убедиться в том, что для определенных параметров  $t_1$  и  $t_2$ , полученных при решении задачи оптимизации, масса потраченного топлива минимальна. Исходя из этого можно прийти к выводу, что задача оптимизации решена, верно.

В таблице 3 представлены результаты расчёта производных массы полезного груза по массе конструкции и скорости истечения газов.

Таблица № 3 - Результаты расчёта производных массы полезного груза

	P = 10.3 кН H = 150 км	P = 8.16 кН H = 150 км	P = 10.3 кН H = 318 км	P = 8.16 кН H = 200 км
$\frac{\partial m_{пг}}{\partial m_{конст}}$ , кг	-0.493651	-0.518138	-0.506582	-0.516467
$\frac{\partial m_{пг}}{\partial w_{ист}}$ , кг	0.294527	0.2937	0.297582	0.294232

Смысл расчета производных массы полезного груза в том, что величина и знак производной наглядно покажет, как изменится масса полезного груза при изменении скорости истечения или массы конструкции.

Производная массы полезного груза по массе конструкции имеет отрицательный знак, это означает, что при увеличении массы конструкции – уменьшится масса полезного груза.

При одной и той же тяге траектория с большей высотой имеет большие абсолютные значения производной как по массе конструкции, так и по скорости истечения.

При одинаковой высоте орбиты  $H = 150$  км, траектория с тягой  $P = 10.3$  кН имеет меньшее абсолютное значение производной по массе полезного груза, а производные по скорости истечения остаются почти равными.

В таблице 4 представлены результаты расчёта потерей скоростей.

Таблица № 4 – Результаты расчёта потерей скоростей

	Параметр	P = 8160Н, h = 200 км	P = 10300Н, h = 318 км	P = 8160 Н, h = 150 км	P = 10300Н, h = 150 км
1-ый АУТ	$V_{ц}, \text{м/с}$	2003.987	1961.283	1755.823	1484.057
	$\Delta V_{упр}, \text{м/с}$	25.975	64.972	15.425	36.526
	$\Delta V_{срав}, \text{м/с}$	543.483	481.628	514.696	411.038

	$V_k, \text{ м/с}$	1434.529	1414.682	1225.703	1036.493
ПУТ	$V_u, \text{ м/с}$	1434.529	1414.682	1225.703	1036.493
	$\Delta V_{упр}, \text{ м/с}$	0.0	0.0	0.0	0.0
	$\Delta V_{грав}, \text{ м/с}$	68.738	171.037	46.552	81.571
	$V_k, \text{ м/с}$	1365.766	1243.588	1179.138	954.893
2-й АУТ	$V_u, \text{ м/с}$	1590.697	1544.414	1612.421	1613.841
	$\Delta V_{упр}, \text{ м/с}$	0.019	0.026	0.151	0.442
	$\Delta V_{грав}, \text{ м/с}$	0.104	0.132	0.772	1.903
	$V_k, \text{ м/с}$	1590.574	1544.256	1611.497	1611.496

На первом АУТ наибольшие потери на управление имеют траектории с большей тягой, это можно объяснить тем, что интеграл для расчета этого вида потери в большей мере зависит от величины тяги, а именно, чем больше тяга, тем больше потеря на управление. Для гравитационных потерь наблюдается обратная тенденция, так как чем больше тяга, тем быстрее аппарат набирает скорость и удаляется от поверхности Луны, уменьшается влияние гравитации, следовательно уменьшаются гравитационные потери.

На ПУТ двигатель выключен, тяга равна нулю, следовательно потери на управление равны нулю.

При одинаковой высоте орбиты траектория с большей тягой имеет меньшие гравитационные потери, так как СВ движется по траектории с большей крутизной, а значит быстрее удаляется от поверхности Луны и быстрее ослабевают гравитационные сил.

На втором АУТ можно заметить, что на всех траекториях гравитационные потери и потери на управление малы. Это можно объяснить тем, что на этом участке аппарат имеет малые углы атаки и угол наклона вектора скорости.

В таблицах 5 – 8 представлены элементы траекторий, полученные в результате интегрирования системы ДУ, для параметров, полученных в результате решения задачи оптимизации с шагом 50 секунд.

Таблица № 5 - Параметры траектории, при P = 8160Н, h = 200км.

t, с	m, кг	Vx, м/с	Vy, м/с	x, км	y, км	θс, град	θс, град	θ град	α,град	h,км	V,м/с	r,км	θ,град	φ,град
0.000	3000.000	0.000	0.000	0.000	0.000	90.000	90.000	90.000	0.000	0.000	0.000	1738.000	90.000	0.000
14.000	2966.101	0.000	15.573	0.000	0.109	90.000	90.000	90.000	0.000	0.109	15.573	1738.109	90.000	0.000
50.000	2878.932	6.438	57.431	0.077	1.416	82.686	83.604	83.606	-0.917	1.416	57.791	1739.416	82.689	0.003
100.000	2757.864	37.539	117.792	1.067	5.792	72.528	72.323	72.359	0.205	5.793	123.629	1743.793	72.564	0.035
150.000	2636.795	95.462	176.987	4.278	13.176	62.371	61.659	61.799	0.712	13.181	201.091	1751.181	62.511	0.140
200.000	2515.727	180.704	230.539	11.069	23.397	52.213	51.910	52.270	0.303	23.432	292.920	1761.432	52.573	0.360
250.000	250.000	2394.659	292.987	273.663	22.800	36.056	42.055	43.047	8160.000	1.558	36.203	400.915	1774.203	42.791
300.000	2273.591	431.207	301.343	40.800	50.506	31.897	34.947	36.254	-3.050	50.972	526.068	1788.972	33.204	1.307
350.000	2152.522	593.399	308.408	66.321	65.847	21.739	27.462	29.568	-5.723	67.066	668.759	1805.066	23.845	2.106
400.000	2031.454	776.703	289.629	100.492	80.916	11.581	20.450	23.613	-8.869	83.690	828.947	1821.690	14.744	3.162
450.000	1910.386	977.339	239.812	144.280	94.292	1.423	13.786	18.289	-12.363	99.964	1006.330	1837.964	5.926	4.502
500.000	1789.318	1190.581	153.888	198.437	104.296	-8.735	7.365	13.513	-16.099	114.952	1200.485	1852.952	-2.587	6.148
550.000	1668.249	1410.729	27.004	263.454	108.998	-18.892	1.097	9.214	-19.989	127.693	1410.988	1865.693	-10.775	8.118
555.370	1655.246	1434.514	10.747	271.094	109.100	-19.983	0.429	8.779	-20.413	128.888	1434.554	1866.888	-11.634	8.350
600.000	1655.246	1424.403	-50.891	334.898	108.200	-19.983	-2.046	8.235	-17.937	138.330	1425.312	1876.330	-9.702	10.282
650.000	1655.246	1410.769	-118.798	405.787	103.953	-19.983	-4.813	7.611	-15.170	148.122	1415.762	1886.122	-7.559	12.424
700.000	1655.246	1394.778	-185.485	475.935	96.341	-19.983	-7.575	6.970	-12.408	157.078	1407.057	1895.078	-5.438	14.545
750.000	1655.246	1376.506	-250.944	545.227	85.425	-19.983	-10.332	6.315	-9.652	165.195	1399.193	1903.195	-3.336	16.647
800.000	1655.246	1356.021	-315.171	613.549	71.267	-19.983	-13.085	5.648	-6.899	172.469	1392.166	1910.469	-1.251	18.733
850.000	1655.246	1333.384	-378.159	680.793	53.929	-19.983	-15.834	4.969	-4.150	178.895	1385.972	1916.895	0.819	20.803
900.000	1655.246	1308.649	-439.903	746.852	33.472	-19.983	-18.580	4.280	-1.403	184.473	1380.607	1922.473	2.877	22.860
950.000	1655.246	1281.862	-500.397	811.624	9.959	-19.983	-21.324	3.583	1.341	189.199	1376.070	1927.199	4.923	24.907
1000.000	1655.246	1253.066	-559.634	875.005	-16.547	-19.983	-24.066	2.878	4.083	193.071	1372.358	1931.071	6.961	26.944
1050.000	1655.246	1222.296	-617.604	936.897	-45.983	-19.983	-26.807	2.167	6.823	196.088	1369.468	1934.088	8.991	28.974
1100.000	1655.246	1189.583	-674.297	997.202	-78.286	-19.983	-29.546	1.453	9.563	198.250	1367.400	1936.250	11.015	30.999
1150.000	1655.246	1154.950	-729.701	1055.824	-113.391	-19.983	-32.285	0.735	12.301	199.554	1366.153	1937.554	13.036	33.020
1175.420	1655.246	1136.614	-757.368	1084.950	-132.292	-35.240	-33.677	0.369	-1.563	199.889	1365.833	1937.889	-1.194	34.046
1200.000	1595.728	1219.205	-855.003	1113.897	-152.104	-35.240	-35.041	0.042	-0.199	199.997	1489.124	1937.997	-0.157	35.083
1219.544	1548.403	1287.243	-934.308	1138.387	-169.587	-35.240	-35.973	0.000	0.733	200.000	1590.575	1938.000	0.733	35.973

Таблица №6 - Параметры траектории, при Р = 10300Н, h = 318км

t, с	m, кг	Vx, м/с	Vy, м/с	x, км	y, км	θс, град	θс, град	θ град	α,град	h,км	V,м/с	r,км	θ,град	φ,град
0.000	3000.000	0.000	0.000	0.000	0.000	90.000	90.000	90.000	0.000	0.000	0.000	1738.000	90.000	0.000
14.000	2957.211	0.000	25.690	0.000	0.179	90.000	90.000	90.000	0.000	0.179	25.690	1738.179	90.000	0.000
50.000	2847.181	10.326	94.551	0.123	2.333	80.777	83.768	83.772	-2.990	2.333	95.113	1740.333	80.781	0.004
100.000	2694.362	60.455	192.524	1.715	9.513	67.968	72.567	72.623	-4.599	9.514	201.793	1747.514	68.024	0.056
150.000	2541.543	153.971	285.330	6.893	21.500	55.159	61.648	61.872	-6.489	21.514	324.223	1759.514	55.383	0.224
200.000	2388.724	291.123	363.490	17.840	37.802	42.349	51.308	51.884	-8.959	37.892	465.701	1775.892	42.925	0.576
250.000	2235.905	470.145	416.899	36.704	57.437	29.540	41.565	42.736	-12.025	57.812	628.364	1795.812	30.711	1.171
300.000	2083.086	687.124	435.087	65.489	78.905	16.731	32.342	34.406	-15.611	80.085	813.289	1818.085	18.795	2.064
350.000	1930.267	935.887	407.504	105.948	100.183	3.921	23.529	26.828	-19.608	103.233	1020.757	1841.233	7.220	3.299
400.000	1777.448	1207.919	323.806	159.467	118.720	-8.888	15.006	19.915	-23.894	125.556	1250.567	1863.556	-3.979	4.909
433.059	1676.406	1395.327	232.811	202.488	128.004	-17.357	9.473	15.666	-26.830	138.958	1414.616	1876.958	-11.164	6.193
500.000	1676.406	1383.222	141.681	295.509	140.522	-17.357	5.848	14.788	-23.206	163.623	1390.459	1901.623	-8.418	8.940
550.000	1676.406	1371.622	75.515	364.389	145.945	-17.357	3.151	14.098	-20.509	180.861	1373.699	1918.861	-6.411	10.947
600.000	1676.406	1357.973	10.929	432.637	148.100	-17.357	0.461	13.380	-17.819	197.083	1358.017	1935.083	-4.438	12.919
650.000	1676.406	1342.382	-52.121	500.154	147.064	-17.357	-2.224	12.636	-15.134	212.287	1343.393	1950.287	-2.498	14.860
700.000	1676.406	1324.946	-113.677	566.844	142.913	-17.357	-4.904	11.867	-12.454	226.471	1329.813	1964.471	-0.586	16.771
750.000	1676.406	1305.749	-173.780	632.619	135.720	-17.357	-7.581	11.075	-9.777	239.633	1317.262	1977.633	1.299	18.656
800.000	1676.406	1284.867	-232.466	697.391	125.558	-17.357	-10.255	10.262	-7.102	251.775	1305.727	1989.775	3.160	20.517
850.000	1676.406	1262.366	-289.774	761.079	112.497	-17.357	-12.928	9.428	-4.429	262.894	1295.197	2000.894	4.999	22.357
900.000	1676.406	1238.303	-345.738	823.602	96.603	-17.357	-15.600	8.577	-1.758	272.992	1285.662	2010.992	6.819	24.177
950.000	1676.406	1212.728	-400.389	884.884	77.945	-17.357	-18.271	7.708	0.913	282.068	1277.114	2020.068	8.622	25.979
1000.000	1676.406	1185.684	-453.758	944.850	56.586	-17.357	-20.942	6.825	3.584	290.122	1269.544	2028.122	10.409	27.767
1050.000	1676.406	1157.207	-505.871	1003.428	32.590	-17.357	-23.612	5.929	6.255	297.155	1262.946	2035.155	12.184	29.541
1100.000	1676.406	1127.327	-556.754	1060.547	6.019	-17.357	-26.283	5.021	8.926	303.167	1257.315	2041.167	13.947	31.304
1150.000	1676.406	1096.070	-606.427	1116.138	-23.065	-17.357	-28.955	4.103	11.597	308.159	1252.646	2046.159	15.700	33.057
1200.000	1676.406	1063.453	-654.910	1170.132	-54.604	-17.357	-31.626	3.177	14.269	312.130	1248.935	2050.130	17.446	34.803
1250.000	1676.406	1029.492	-702.218	1222.461	-88.537	-17.357	-34.298	2.245	16.941	315.081	1246.180	2053.081	19.186	36.543
1300.000	1676.406	994.196	-748.365	1273.059	-124.806	-17.357	-36.970	1.309	19.613	317.012	1244.378	2055.012	20.921	38.279
1342.682	1676.406	963.014	-786.846	1314.831	-157.572	-40.834	-39.251	0.508	-1.583	317.853	1243.593	2055.853	-1.075	39.759

Продолжение таблицы № 6

1350.000	1654.039	991.818	-822.958	1321.984	-163.462	-40.834	0.333	-1.150	1.160	317.921	1288.784	2055.921	-0.817	40.017
1389.521	1533.248	1155.237	-1024.771	1364.365	-199.935	-40.834	0.000	0.741	1.160	318.000	1544.256	2056.000	0.741	41.575

Таблица №7 - Параметры траектории, при  $P = 8160H$ ,  $h = 150km$ .

t, с	m, кг	Vx, м/с	Vy, м/с	x, км	y, км	θс, град	θс, град	θ град	α,град	h,км	V,м/с	r,км	θ,град	φ,град
0.000	3000.000	0.000	0.000	0.000	0.000	90.000	90.000	90.000	0.000	0.000	0.000	1738.000	90.000	0.000
14.000	2966.101	0.000	15.573	0.000	0.109	90.000	90.000	90.000	0.000	0.109	15.573	1738.109	90.000	0.000
50.000	2878.932	6.685	57.410	0.080	1.416	82.406	83.359	83.361	-0.953	1.416	57.797	1739.416	82.408	0.003
100.000	2757.864	38.956	117.492	1.107	5.786	71.858	71.656	71.693	0.201	5.786	123.782	1743.786	71.894	0.036
150.000	2636.795	98.972	175.777	4.438	13.135	61.310	60.618	60.763	0.692	13.141	201.725	1751.141	61.455	0.145
200.000	2515.727	187.091	227.399	11.473	23.253	50.763	50.554	50.928	0.208	23.290	294.471	1761.290	51.136	0.373
250.000	2394.659	302.782	267.192	23.607	35.678	40.215	41.427	42.190	-1.212	35.835	403.818	1773.835	40.977	0.763
300.000	2273.591	444.580	289.787	42.187	49.685	29.667	33.097	34.449	-3.430	50.183	530.687	1788.183	31.019	1.352
350.000	2152.522	610.042	289.717	68.460	64.279	19.119	25.404	27.579	-6.284	65.579	675.342	1803.579	21.295	2.175
400.000	2031.454	795.720	261.534	103.529	78.189	8.572	18.194	21.457	-9.623	81.137	837.598	1819.137	11.834	3.263
450.000	1910.386	997.144	199.933	148.295	89.875	-1.976	11.338	15.976	-13.314	95.881	1016.990	1833.881	2.662	4.638
500.000	1789.318	1208.803	99.870	203.414	97.541	-12.524	4.723	11.047	-17.247	108.777	1212.922	1846.777	-6.200	6.324
503.125	1781.751	1222.228	92.230	207.213	97.841	-13.183	4.315	10.755	-17.498	109.498	1225.703	1847.498	-6.743	6.440
550.000	1781.751	1213.708	25.814	264.313	100.604	-13.183	1.218	9.399	-14.401	119.505	1213.982	1857.505	-5.002	8.181
600.000	1781.751	1202.528	-43.987	324.727	100.145	-13.183	-2.095	7.924	-11.088	128.608	1203.332	1866.608	-3.164	10.019
650.000	1781.751	1189.244	-112.782	384.530	96.222	-13.183	-5.417	6.423	-7.766	136.095	1194.579	1874.095	-1.343	11.840
700.000	1781.751	1173.897	-180.636	443.617	88.882	-13.183	-8.748	4.901	-4.435	141.972	1187.713	1879.972	0.466	13.649
750.000	1781.751	1156.516	-247.611	501.886	78.173	-13.183	-12.085	3.363	-1.098	146.243	1182.725	1884.243	2.265	15.448
800.000	1781.751	1137.116	-313.764	559.235	64.135	-13.183	-15.426	1.814	2.243	148.911	1179.610	1886.911	4.057	17.240
811.547	1781.751	1132.349	-328.931	572.339	60.424	-19.398	-16.198	1.456	-3.200	149.300	1179.157	1887.300	-1.745	17.653
850.000	1688.643	1286.275	-439.212	618.815	45.665	-19.398	-18.853	0.280	-0.545	149.959	1359.195	1887.959	-0.265	19.133
900.000	1567.575	1498.980	-587.010	688.383	20.031	-19.398	-21.386	-0.002	1.988	150.000	1609.820	1888.000	1.985	21.384
900.323	1566.794	1500.402	-587.981	688.867	19.841	-19.398	-21.399	0.000	2.001	150.000	1611.499	1888.000	2.001	21.399



Таблица № 8 - Параметры траектории, при  $P = 10300H$ ,  $h = 150km$ .

t, с	m, кг	V <sub>x</sub> , м/с	V <sub>y</sub> , м/с	x, км	y, км	θс, град	θс, град	θ град	α,град	h,км	V,м/с	r,км	θ,град	φ,град
0.000	3000.000	0.000	0.000	0.000	0.000	90.000	90.000	90.000	0.000	0.000	0.000	1738.000	90.000	0.000
14.000	2957.211	0.000	25.690	0.000	0.179	90.000	90.000	90.000	0.000	0.179	25.690	1738.179	90.000	0.000
50.000	2847.181	11.837	94.375	0.141	2.332	79.421	82.851	82.856	-3.431	2.332	95.115	1740.332	79.425	0.005
100.000	2694.362	69.074	190.067	1.963	9.461	64.727	70.028	70.092	-5.301	9.462	202.229	1747.462	64.791	0.064
150.000	2541.543	174.860	275.438	7.859	21.167	50.034	57.591	57.847	-7.557	21.185	326.255	1759.185	50.290	0.256
200.000	2388.724	327.647	337.991	20.234	36.624	35.340	45.890	46.543	-10.550	36.740	470.734	1774.740	35.993	0.653
250.000	2235.905	522.698	364.955	41.330	54.373	20.646	34.923	36.244	-14.277	54.849	637.499	1792.849	21.967	1.321
300.000	2083.086	751.976	343.905	73.076	72.319	5.953	24.576	26.888	-18.623	73.793	826.885	1811.793	8.264	2.312
349.623	1931.419	1002.192	264.271	116.533	87.672	-8.630	14.772	18.424	-23.402	91.388	1036.449	1829.388	-4.978	3.652
350.000	1931.419	1002.156	263.720	116.911	87.772	-8.630	14.743	18.407	-23.373	91.511	1036.275	1829.511	-4.966	3.664
400.000	1931.419	996.551	191.315	166.886	99.142	-8.630	10.867	16.058	-19.497	106.706	1014.749	1844.706	-3.439	5.191
450.000	1931.419	989.146	120.175	216.536	106.924	-8.630	6.927	13.621	-15.557	119.588	996.419	1857.588	-1.936	6.694
500.000	1931.419	980.005	50.144	265.772	111.178	-8.630	2.929	11.108	-11.559	130.179	981.287	1868.179	-0.451	8.179
550.000	1931.419	969.172	-18.926	314.508	111.955	-8.630	-1.119	8.530	-7.511	138.499	969.357	1876.499	1.019	9.649
600.000	1931.419	956.674	-87.171	362.661	109.299	-8.630	-5.206	5.901	-3.424	144.561	960.637	1882.561	2.477	11.107
650.000	1931.419	942.522	-154.722	410.148	103.249	-8.630	-9.322	3.236	0.692	148.377	955.137	1886.377	3.928	12.558
653.299	1931.419	941.530	-159.157	413.256	102.731	-14.251	-9.595	3.059	-4.657	148.550	954.888	1886.550	-1.598	12.653
700.000	1788.681	1177.420	-285.381	462.666	92.368	-14.251	-13.625	0.561	-0.627	149.938	1211.511	1887.938	-0.066	14.186
750.000	1635.862	1451.121	-425.841	528.281	74.613	-14.251	-16.355	-0.106	2.103	150.027	1512.314	1888.027	1.997	16.249
765.533	1588.387	1541.209	-470.747	551.518	67.650	-14.251	-16.985	0.000	2.734	150.000	1611.499	1888.000	2.733	16.985

В таблице 9 приведены параметры промежуточных орбит, сформированных после конца первого АУТ.

Таблица № 9 - Параметры промежуточных орбит

Параметры орбиты	$P = 8160H,$ $h = 200 \text{ км}$	$P = 10300H,$ $h = 318 \text{ км}$	$P = 8160 H,$ $h = 150 \text{ км}$	$P = 10300H,$ $h = 150 \text{ км}$
$a, \text{км}$	1534.721	1521.216	1288.442	1143.974
$e, [-]$	0.263	0.352	0.465	0.651
$r_{п}, \text{км}$	1131.438	986.429	688.884	399.947
$r_{а}, \text{км}$	1938.003	2056.002	1888.000	1888.000

Для орбит с равной высотой, при меньшей тяге после конца первого АУТ необходимо сформировать промежуточную орбиту с меньшим эксцентриситетом, чтобы радиус перицентра увеличился. Тогда во время второго АУТ необходимо будет сообщить СВ меньшее приращение скорости для перехода на круговую орбиту.

Промежуточные орбиты, сформированные после конца первого АУТ имеют  $r_a$  равный радиусу  $r_{ц}$  целевой орбиты, потому что за первый АУТ СВ сообщается ровно та скорость, которая необходима, чтобы радиус апоцентра промежуточной орбиты, по которой движется СВ после окончания первого АУТ, был равен радиусу целевой орбиты.

На рисунках 3 – 9 представлены графики зависимости  $y(x)$ ,  $\theta(t)$ ,  $\theta_c(t)$ ,  $\vartheta_c(t)$ ,  $\alpha(t)$ ,  $V(t)$ ,  $h(t)$  соответственно.

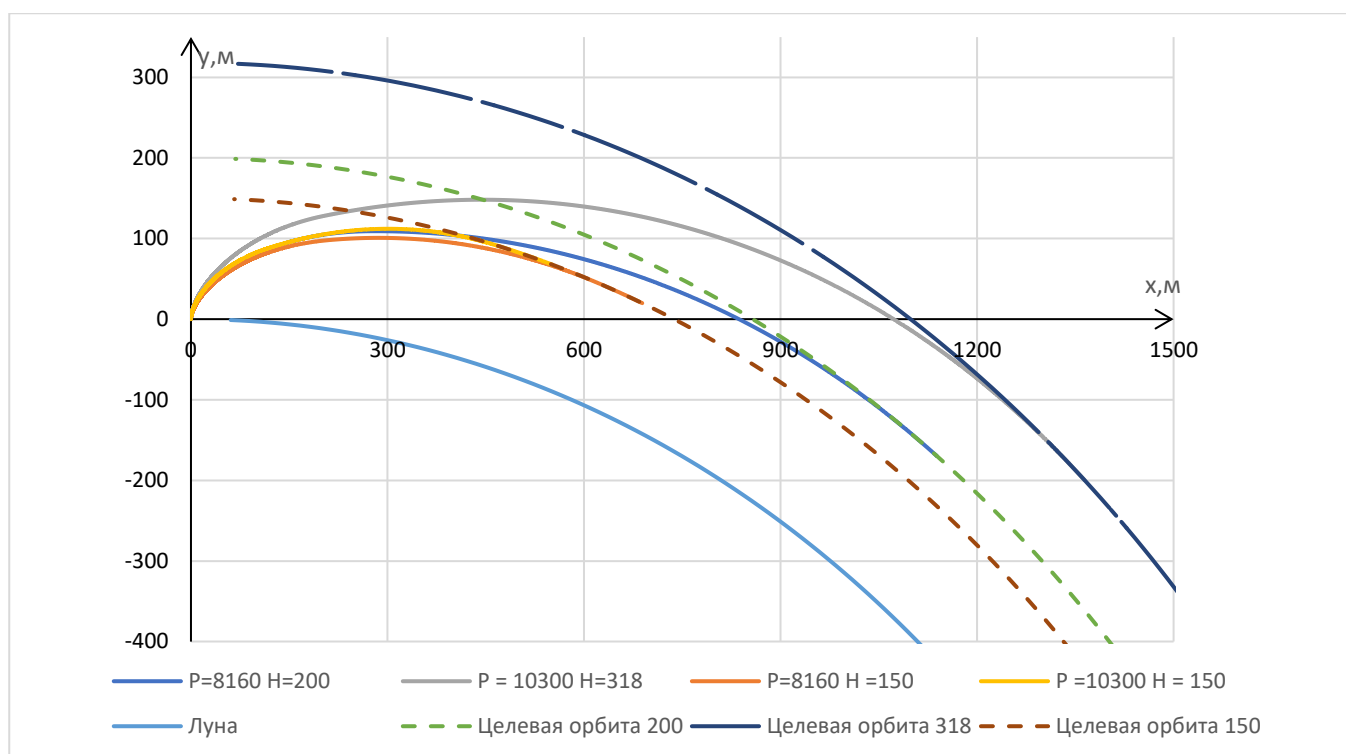


Рисунок 3 – Графики траекторий полёта, для различных начальных данных

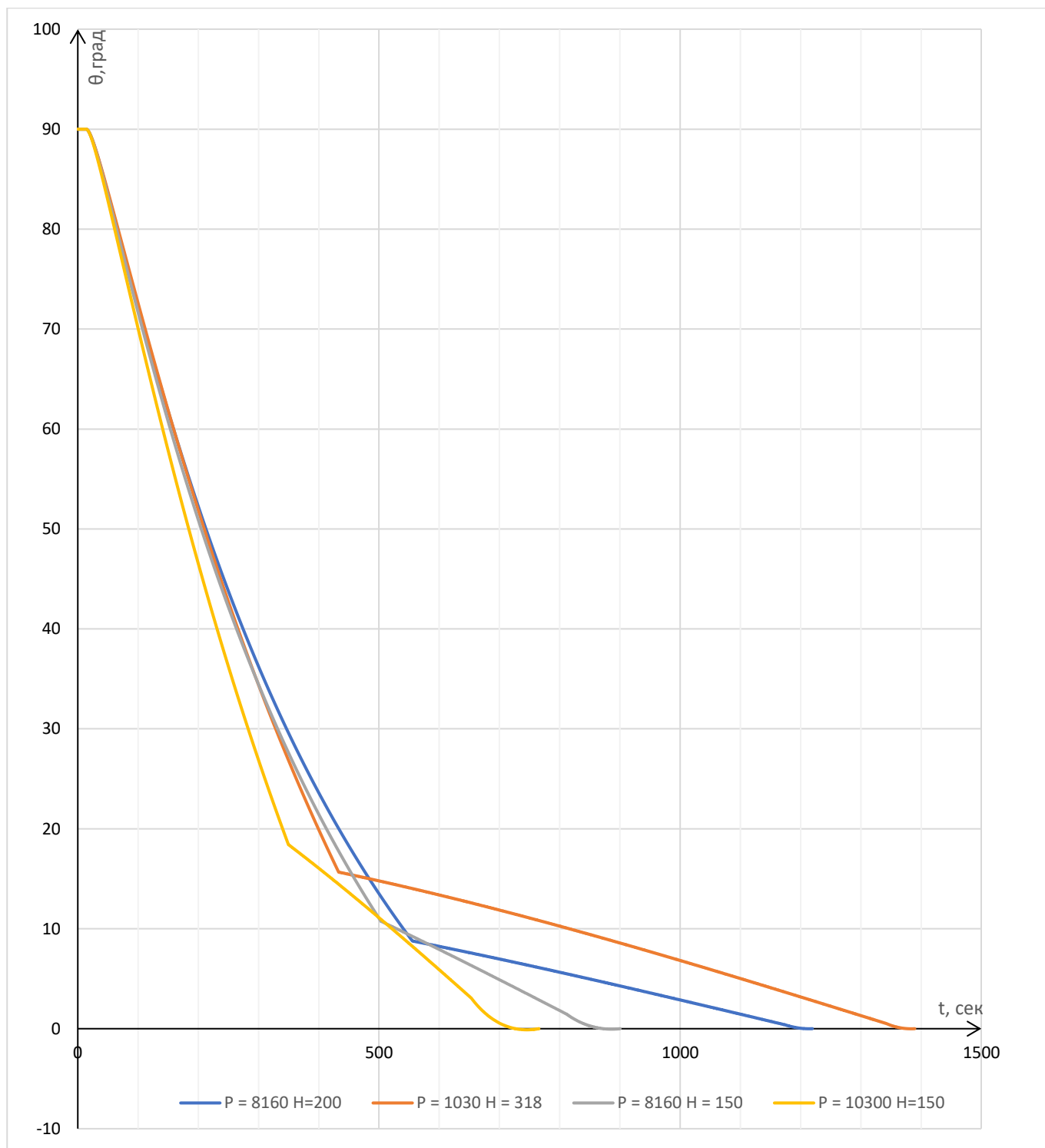


Рисунок 4 – Графики зависимости угла наклона вектора скорости от времени

На первом участке до 14 секунд полета происходит вертикальный старт СВ, ему соответствует горизонтальный короткий участок на графике. На первом АУТ угол наклона вектора скорости уменьшается быстрее всего. В конце первого АУТ на графике виден излом, это связано с отключением двигательной установки. Излом можно зафиксировать на графике тем раньше, чем больше тяга, так как первый АУТ

меньше. Если анализировать траектории с одинаковой тягой, то СВ поворачивает быстрее, если аппарат выводится на более низкую орбиту. Интенсивность изменения угла наклона вектора скорости на ПУТ заметно снижается, так как двигатель выключен. В конце ПУТ угол наклона вектора скорости принимает значение близкое к нулю. Это необходимо, **для того чтобы вектор тяги был направлен по направлению, близкому к касательной к траектории**. В этот момент СВ находится в точке близкой к апоселению эллиптической орбиты. На втором АУТ угол наклона вектора скорости стремится к нулю (одно из условий выхода СВ на целевую круговую орбиту).

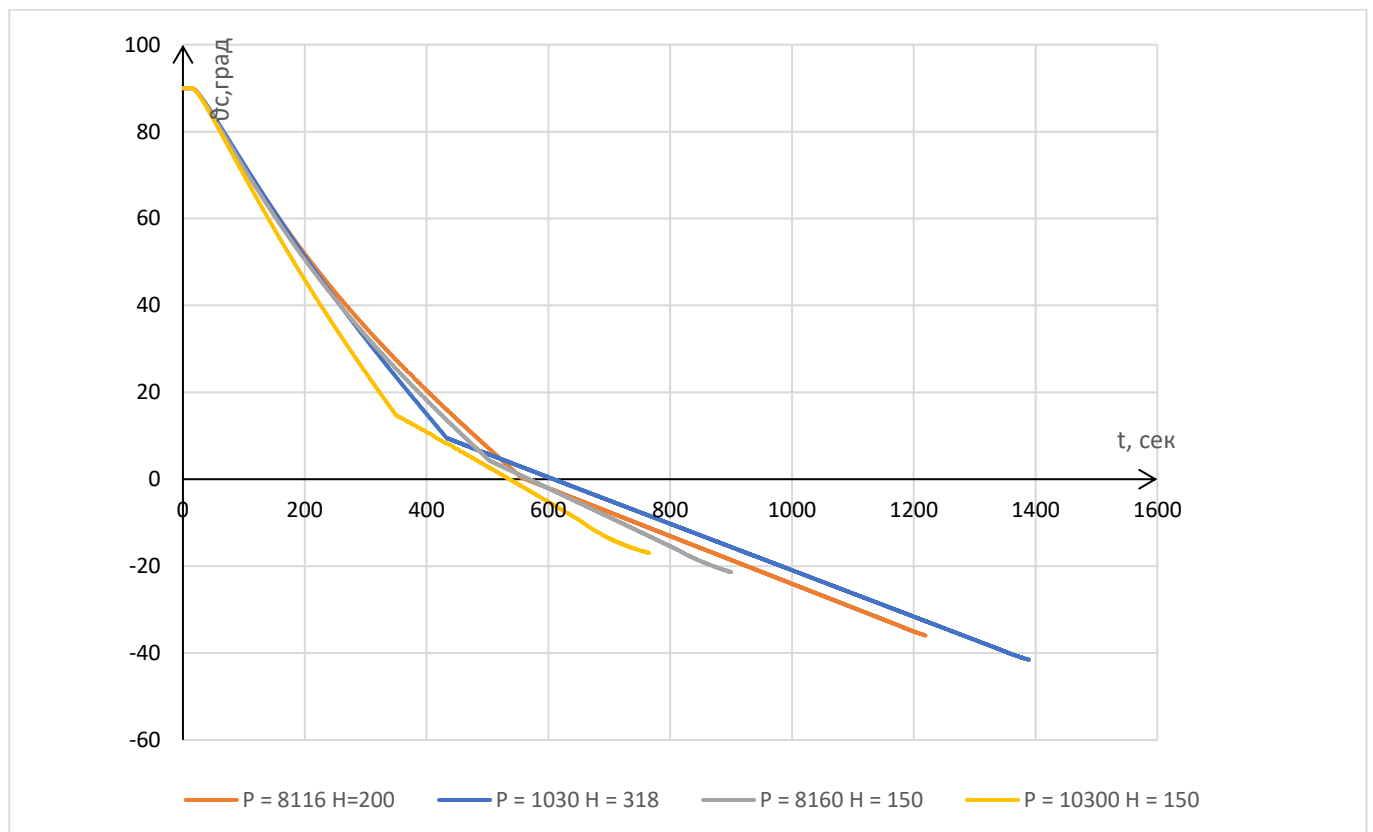


Рисунок 5 – Графики зависимости угла наклона вектора скорости в стартовой СК от времени

Характер изменения угла наклона вектора скорости в стартовой СК  $\theta_c$  аналогичен изменению угла  $\theta$  (рисунок 4). Отрицательные значения объясняются взаимным расположением местного и стартового горизонта.

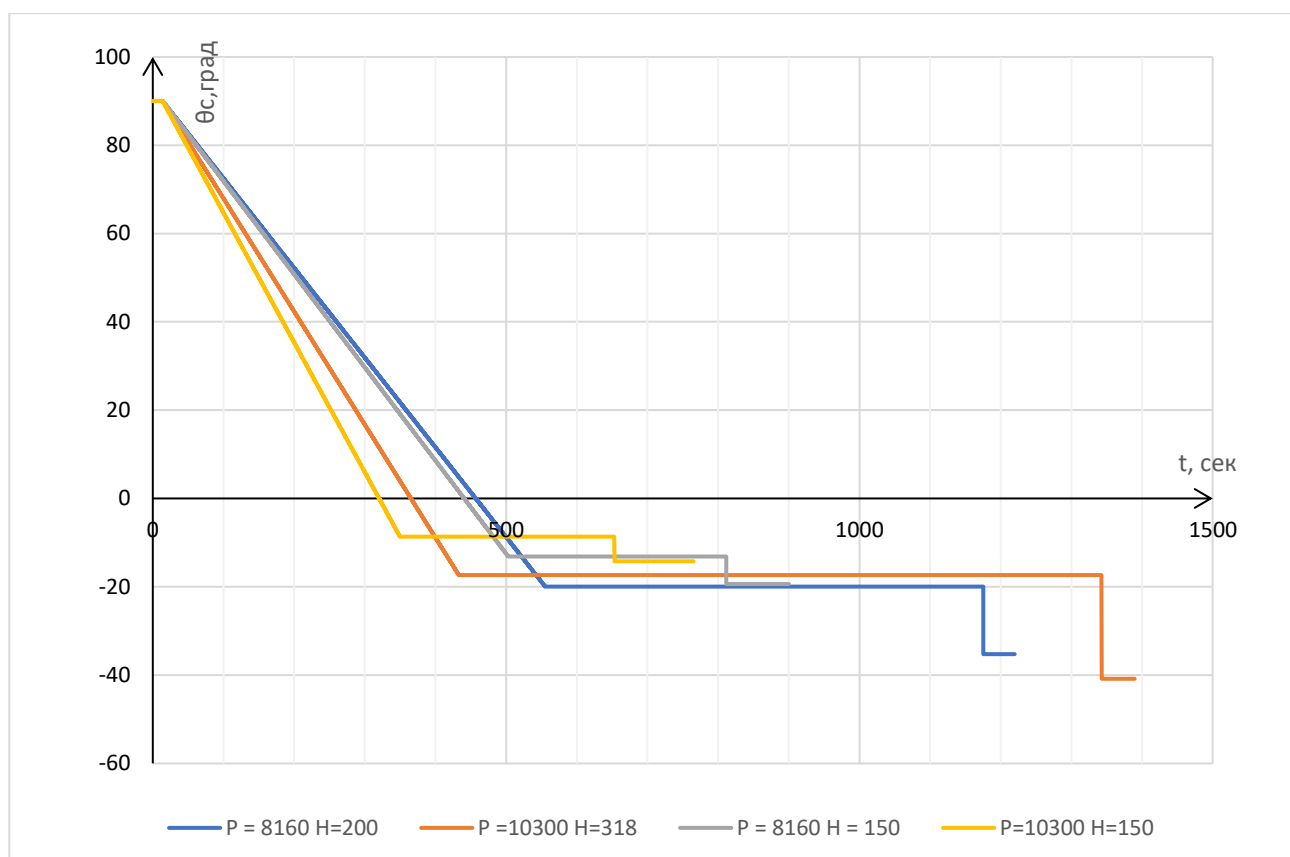


Рисунок 6 – Графики зависимости тангажа в стартовой системе координат от времени

При вертикальном старте 14 секунд угол тангажа остаётся равным  $90^\circ$ . На первом АУТ тангаж меняется по линейному закону, согласно закону изменения угла тангажа(1.2). При равных значениях тяги угол тангажа изменяется быстрее в случае меньшей высоты орбиты, так как при большей тяге уменьшается время полета. В случае выведения с одинаковой тягой, у орбит с большей высотой увеличивается время полёта, следовательно угол тангажа изменяется быстрее при выведении на орбиты с меньшей высотой. На ПУТ не рассматривается изменение угла тангажа ввиду принятой программы тангажа. На втором АУТ угол тангажа в стартовой СК на всех траектория остаётся постоянным.

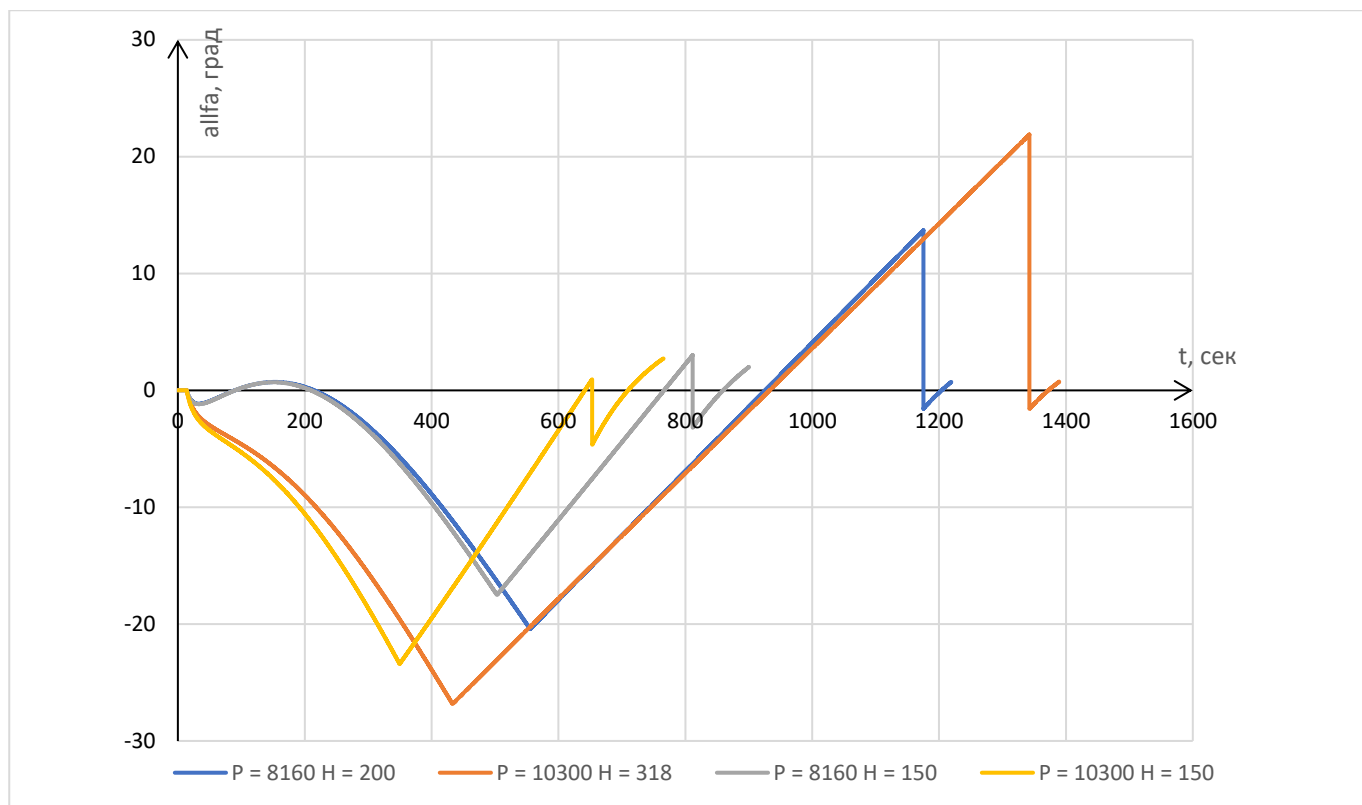


Рисунок 7 – Графики зависимости угла атаки от времени

На первом АУТ при выведении с одинаковой тягой характер изменения угла атаки практически одинаков, но стоит отметить, что в случае высоких орбит угол атаки уменьшается до большего значения по модулю, так как длительность АУТ больше. При равной высоте орбиты угол атаки изменяется быстрее в случае большей тяги. На ПУТ угол атаки не определен, ввиду выбранной программой тангажа. На втором АУТ угол атаки увеличивается во всех случаях с одинаковой скоростью, меняя знак, но принимая значения близкие к нулю для минимизации потерь на управление и направления вектора тяги на втором АУТ по касательной к траектории.

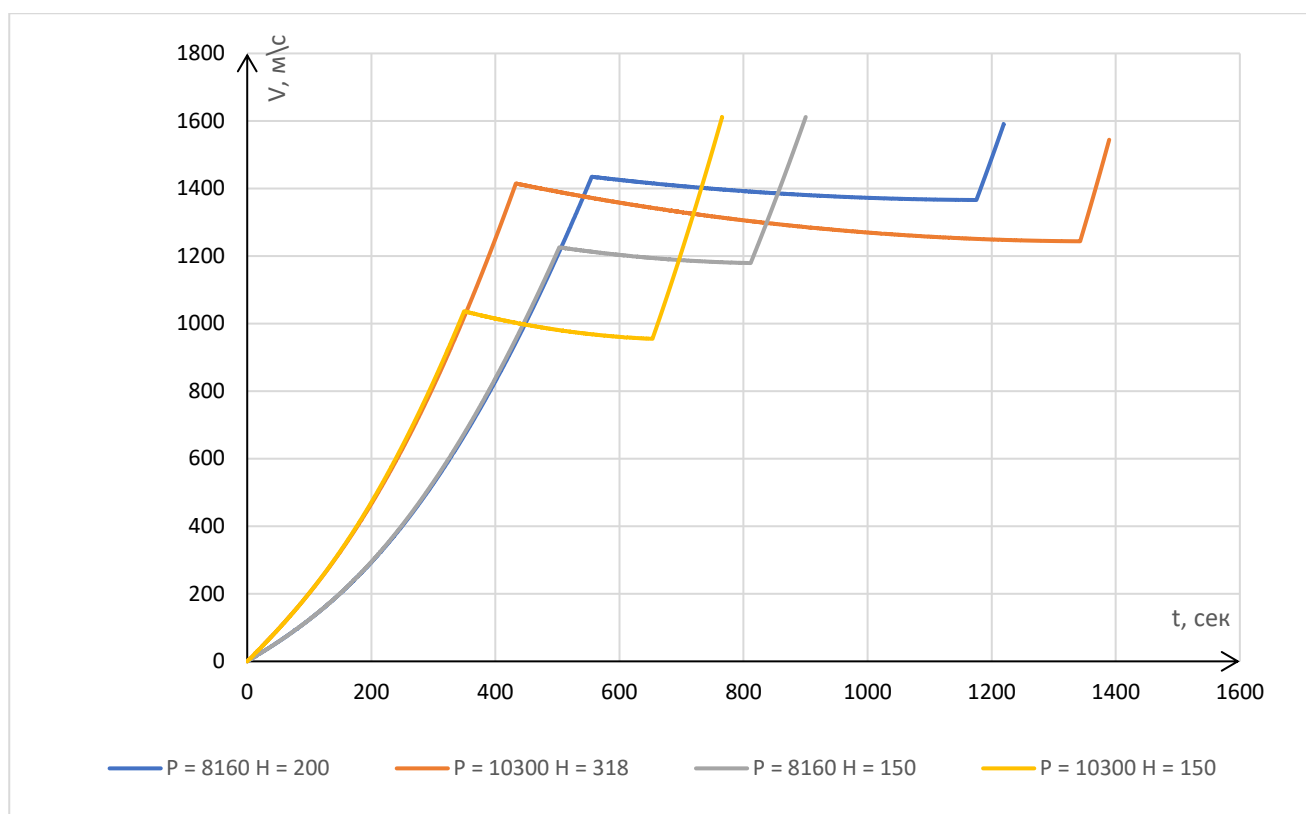


Рисунок 8 – Графики зависимости скорости от времени

На первом АУТ скорость при равной тяге и разной высоте орбиты меняется одинаково, но для орбит с большей высотой увеличивается до большего значения, так как для достижения более высоких орбит необходимо набрать большую скорость на первой АУТ. Также при равной высоте выведения скорость достигает большего значения в конце первого АУТ для траекторий с меньшей тягой, чтобы в дальнейшем на втором АУТ потребовалось меньшее приращение скорости для выхода на заданную орбиту. На ПУТ скорость уменьшается, так как двигатель не работает, но уменьшается медленно, так как есть только гравитационная составляющая действующих сил. На втором АУТ скорости увеличиваются до значений, необходимых для достижения заданной орбиты.



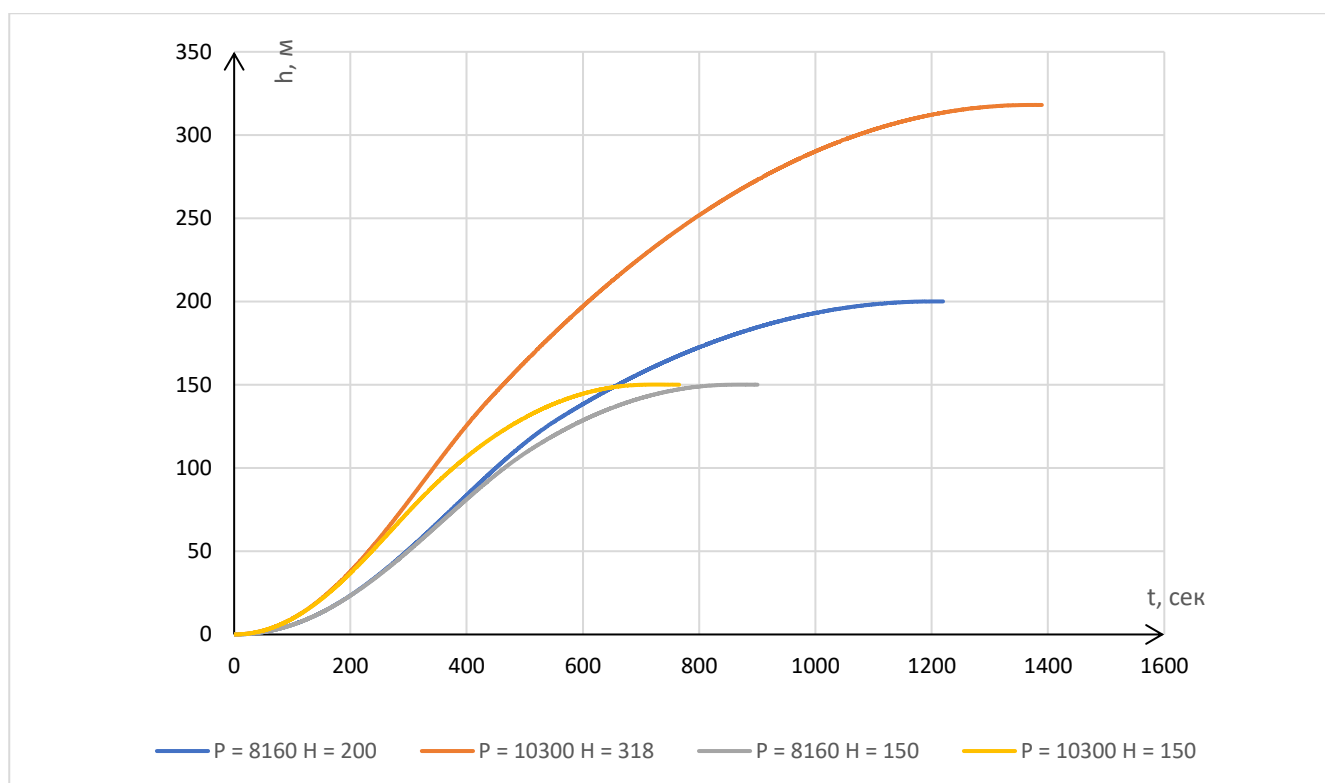


Рисунок 9 – Графики зависимости высоты орбиты от времени

На первом АУТ высота орбиты растёт стремительнее чем на ПУТ, это объясняется тем, что работает двигательная установка. По данным графика основной набор высоты происходит на ПУТ. На втором АУТ высота орбиты почти не изменяется, так как на нём происходит только набор необходимой скорости и разворот на необходимый угол наклона вектора скорости к местному горизонту.

При равной высоте орбиты СВ выводится быстрее в случае, когда тяга больше.

## Выводы

В ходе выполнения курсовой работы были освоены методы определения параметров программы управления средством выведения (СВ) при выведении полезного груза (ПГ) на целевую орбиту.

Для определения параметров программы управления было разработано программно-математическое обеспечение на языке программирования C++, которое позволило определить параметры программы управления вектором тяги ДУ СВ при компланарном выведении СВ на орбиту ИСЛ с поверхности Луны для четырех траекторий, при этом одним из условий создания этой программы было нахождение оптимизированных параметров, при которых масса топлива была бы минимальной. В состав обеспечения входило множество подпрограмм. В одной из основных таких подпрограмм было осуществлено интегрирование математической модели СВ методом Рунге- Кутты 4 порядка. Следующая подпрограмма решала краевую задачу и находила значения  $\dot{\vartheta}_1, \vartheta_2$ . При создании данной части программы был освоен итерационный метод решения краевых задач – модифицированный метод Ньютона, применяющий формулы численного дифференцирования. Далее решалась задача оптимизации, по результатам которой находились наиболее выгодные параметры управления для минимизации массы топлива. Для решения этой задачи был освоен метод градиентного спуска. По результатам интегрирования был произведен сравнительный анализ влияния параметров управления вектором тяги ДУ на характеристики орбиты. Были построены графики зависимости различных параметров от времени, получены производные массы полезного груза по массе конструкции и скорости истечения газов, посчитаны потери по скорости, а также посчитаны параметры промежуточных орбит. После анализа всех результатов были сделаны следующие выводы

1. Производная массы полезного груза по массе конструкции имеет отрицательный знак, это означает, что при увеличении массы конструкции – уменьшится масса полезного груза. В то время как производная по скорости

истечения газов имеет знак плюс, что говорит о том, что при большей скорости истечения – большая масса полезного груза.

2. В процессе выведения СВ на целевую орбиту существует несколько потерь по скорости, гравитационный потери, возникающие вследствие существования гравитационного поля луны и потери скорости на управление. Причем на ПУТ потери на управление отсутствуют, так как двигатель выключен. Гравитационные потери на втором АУТ уже незначительны, так как на большем удалении от поверхности Луны гравитация ослабевает, но и потери на управление также на порядок меньше относительно 1 АУТ, так как в подынтегральном выражении для расчета потери скорости на управление учувствует угол атаки, который на этом участке траектории равен нулю.

3. Данная схема выведения полезного груза на целевую круговую орбиту является оптимальной по ряду причин:

- Основной набор высоты приходится на пассивный участок траектории, наличие ПУТ дает нам возможность, не затрачивая топлива выйти на промежуточную орбиту.
- Использование двигателя с возможностью многократного включения и выключения обеспечивает наличие 2 АУТ, который необходим для выведения СВ с промежуточной орбиты на целевую круговую, путем набора необходимой скорости и разворота на необходимый угол наклона вектора скорости к местному горизонту.

После полученных результатов нетрудно предположить, если уменьшить длительность первого АУТ на любой из предложенных траекторий, тогда после пассивного участка траектории на втором АУТ нам понадобится большее приращение скорости, а следовательно больше топлива будет потрачено, для того чтобы набрать скорость необходимую для выхода на целевую орбиту.

# Приложение А

## Листинг программы

```
#define _USE_MATH_DEFINES
#include <iostream>
#include <cstdlib>
#include <vector>
#include <cmath>
#include <fstream>
#include <iomanip>
struct parametr
{
    double tetta1;
    double tetta2;
    double t1;
    double t2;
};

std::vector <double> runge_kutt_4(parametr p, bool save);

const double Mm = 4903 * 1e9;
const double Rm = 1738 * 1e3;
double P_0 = 10300.0;
double W_0 = 3370.0;
double betta_0 = P_0 / W_0;
double tmax = 1525 / (P_0 / W_0);
double m_0 = 3000.0;

double gx(double x, double y)
{
    return (Mm * x) / (pow((pow(x, 2) + pow((y + Rm), 2)), 1.5));
}

double gy(double x, double y)
{
    return (Mm * (Rm + y)) / (pow((pow(x, 2) + pow((y + Rm), 2)), 1.5));
}

double P(bool off)
{
    if (off)
    {
        return 0;
    }
    else
    {
        return P_0;
    }
}

double tetta(double t, double tb, double t1, double t2, double tk, parametr p)
{
    if (t < tb)
    {
        return M_PI_2;
    }
    if (tb < t && t <= p.t1)
    {
        return M_PI_2 + p.tetta1 * (t - tb);
    }
    if (p.t1 < t && t < p.t2)
```

```

    {
        return M_PI_2 + p.tetta1 * (t - tb);
    }
    if (p.t2 <= t)
    {
        return (p.tetta2);
    }
}

double betta(bool off) {
    double betta_0 = P_0 / W_0;
    if (off)
    {
        return 0.0;
    }
    else
    {
        return betta_0;
    }
}

std::vector <double> runge_kutt_4(parametr p, bool save)
{
    double k1x, k1y, k1vx, k1vy, k1m, k2x, k2y, k2vx, k2vy, k2m, k3x, k3y, k3vx, k3vy,
    k3m, k4x, k4y, k4vx, k4vy, k4m;
    double x, y, vx, vy, m, m_new, x_new, y_new, vx_new, vy_new, v_new, dt, t;
    double t2, tk, t1, tb;
    double h, t_o, c, dv, l;
    double H, vk, v, mt;
    double r, TEtta, tettas, alfa, TETTA;
    std::vector <double> rezult;
    tk = 820;
    tb = 14;
    t = 0;
    dt = 0.1;
    y = 0;
    x = 0;
    vx = 0;
    vy = 0;
    h = 0;
    vx_new = 0;
    l = 0;
    vy_new = 0;
    x_new = 0;
    y_new = 0;
    m_new = m_0;
    m = m_0;
    t_o = 0;
    H = 318000;
    vk = sqrt(Mm / (H + Rm));
    bool off = false;
    v = sqrt(vx * vx + vy * vy);
    //std::cout <<"PK4 1 " << m<<" " << W_0<<" " << p.tetta1<<" " << p.tetta2;
    while (abs(v - vk) > 0.000001)
    {
        rezult.clear();
        if (p.t1 <= t && t < p.t2)
        {
            off = true;
        }
        else
        {
            off = false;
        }
    }
}

```

```

k1vx = dt * ((P(off) / m) * cos(tetta(t, tb, p.t1, p.t2, tk, p)) - gx(x, y));
k1vy = dt * ((P(off) / m) * sin(tetta(t, tb, p.t1, p.t2, tk, p)) - gy(x, y));
k1x = dt * vx;
k1y = dt * vy;
k1m = -dt * betta(off);

k2vx = dt * ((P(off) / (m + k1m / 2)) * cos(tetta(t + dt / 2, tb, p.t1, p.t2,
tk, p)) - gx(x + k1x / 2, y + k1y / 2));
k2vy = dt * ((P(off) / (m + k1m / 2)) * sin(tetta(t + dt / 2, tb, p.t1, p.t2,
tk, p)) - gy(x + k1x / 2, y + k1y / 2));
k2x = dt * (vx + k1vx / 2);
k2y = dt * (vy + k1vy / 2);
k2m = -dt * betta(off);

k3vx = dt * ((P(off) / (m + k2m / 2)) * cos(tetta(t + dt / 2, tb, p.t1, p.t2,
tk, p)) - gx(x + k2x / 2, y + k2y / 2));
k3vy = dt * ((P(off) / (m + k2m / 2)) * sin(tetta(t + dt / 2, tb, p.t1, p.t2,
tk, p)) - gy(x + k2x / 2, y + k2y / 2));
k3x = dt * (vx + k2vx / 2);
k3y = dt * (vy + k2vy / 2);
k3m = -dt * betta(off);

k4vx = dt * ((P(off) / (m + k3m)) * cos(tetta(t + dt, tb, p.t1, p.t2, tk, p))
- gx(x + k3x, y + k3y));
k4vy = dt * ((P(off) / (m + k3m)) * sin(tetta(t + dt, tb, p.t1, p.t2, tk, p))
- gy(x + k3x, y + k3y));
k4x = dt * (vx + k3vx);
k4y = dt * (vy + k3vy);
k4m = -dt * betta(off);

vx_new = vx_new + (k1vx + 2 * k2vx + 2 * k3vx + k4vx) / 6;
vy_new = vy_new + (k1vy + 2 * k2vy + 2 * k3vy + k4vy) / 6;
x_new = x_new + (k1x + 2 * k2x + 2 * k3x + k4x) / 6;
y_new = y_new + (k1y + 2 * k2y + 2 * k3y + k4y) / 6;
m_new = m_new + (k1m + 2 * k2m + 2 * k3m + k4m) / 6;

if ((sqrt(vx_new * vx_new + vy_new * vy_new) - vk) > 0)
{
    dt /= 10;
    l = l + 1;
    vx_new = vx_new - (k1vx + 2 * k2vx + 2 * k3vx + k4vx) / 6;
    vy_new = vy_new - (k1vy + 2 * k2vy + 2 * k3vy + k4vy) / 6;
    x_new = x_new - (k1x + 2 * k2x + 2 * k3x + k4x) / 6;
    y_new = y_new - (k1y + 2 * k2y + 2 * k3y + k4y) / 6;
    m_new = m_new - (k1m + 2 * k2m + 2 * k3m + k4m) / 6;
    continue;
}
v_new = sqrt(vx_new * vx_new + vy_new * vy_new);

t = t + dt;
x = x_new;
y = y_new;

```

```

vx = vx_new;
vy = vy_new;
m = m_new;
v = v_new;
t_o = 0;

if (l < 0.5)
{
    dt = 0.1;
}

if (((p.t1 - t) < 0.1) && ((p.t1 - t) > 0.00001))
{
    dt = p.t1 - t;
    c = 0.1 - dt;
}
if (abs(p.t1 - t) < 0.00001)
{
    dt = c;
}

if (((p.t2 - t) < 0.1) && ((p.t2 - t) > 0.00001))
{
    dt = p.t2 - t;
    c = 0.1 - dt;
}
if (abs(p.t2 - t) < 0.00001)
{
    dt = c;
}

r = pow(((y + Rm) * (y + Rm) + x * x), 0.5);
TEtta = asin((x * vx + (y + Rm) * vy) / (pow((x * x + (y + Rm) * (y + Rm)),
0.5) * pow(vx * vx + vy * vy, 0.5)));
tettas = atan(vy / vx);
alfa = tetta(t, tb, p.t1, p.t2, tk, p) - tettas;
TETTA = TEtta + alfa;
mt = 3000 - m;

```

```

rezult.push_back(t); //0
rezult.push_back(m); //1
rezult.push_back(vx); //2
rezult.push_back(vy); //3
rezult.push_back(x / 1000); //4
rezult.push_back(y / 1000); //5
rezult.push_back(tettas * 180 / M_PI); //6
rezult.push_back(TETTA * 180 / M_PI); //7
rezult.push_back(TEtta * 180 / M_PI); //8
rezult.push_back(P(off)); //9
rezult.push_back(alfa * 180 / M_PI); //10
rezult.push_back(sqrt(gx(x, y) * gx(x, y) + gy(x, y) * gy(x, y))); //11
rezult.push_back((r - Rm) / 1000); //12
rezult.push_back(v_new); //13
rezult.push_back(r / 1000); //14
rezult.push_back(tetta(t, tb, p.t1, p.t2, tk, p) * 180 / M_PI); //15
rezult.push_back(atan(x / (Rm + y)) * 180 / M_PI); //16
rezult.push_back(3000 - m); //17
rezult.push_back(r); //18
/*std::ofstream file("rezult1030150toch.txt", std::ios_base::app);

```

```

        if (file.is_open() && save)
        {
            file << std::setprecision(15) << result[0];
            file << "\t";
            file << std::setprecision(15) << result[1];
            file << "\t";
            file << std::setprecision(15) << result[2];
            file << "\t";
            file << std::setprecision(15) << result[3];
            file << "\t";
            file << std::setprecision(15) << result[4];
            file << "\t";
            file << std::setprecision(15) << result[5];
            file << "\t";
            file << std::setprecision(15) << result[6];
            file << "\t";
            file << std::setprecision(15) << result[7];
            file << "\t";
            file << std::setprecision(15) << result[8];
            file << "\t";
            file << std::setprecision(15) << result[9];
            file << "\t";
            file << std::setprecision(15) << result[10];
            file << "\t";
            file << std::setprecision(15) << result[11];
            file << "\t";
            file << std::setprecision(15) << result[12];
            file << "\t";
            file << std::setprecision(15) << result[13];
            file << "\t";
            file << std::setprecision(15) << result[14];
            file << "\t";
            file << std::setprecision(15) << result[15];
            file << "\t";
            file << std::setprecision(15) << result[16];
            file << "\n";
            file.close();

            */
            //std::cout << std::setprecision(15) << result[0] << " " <<
            std::setprecision(15) << " " << result[1] << " " << std::setprecision(15) << " " <<
            result[2] << " " << std::setprecision(15) << " " << result[3] << " " <<
            std::setprecision(15) << " " << result[4] << " " << std::setprecision(15) << " " <<
            result[5] << " " << std::setprecision(15) << " " << result[6]-Rm << " " <<
            std::setprecision(15) << " " << std::setprecision(15) << " " << result[7] << " " <<
            std::setprecision(15) << " " << result[8] << " " << result[9] << " " <<
            std::setprecision(15) << " " << result[10] << " " << std::setprecision(15) << " " <<
            result[11] << " " << std::setprecision(15) << " " << std::setprecision(15) << " " <<
            result[12] << "\n";

        }
        return result;
    }
    std::vector <double> metod(parametr p)
    {
        double drob;
        double dtetta1 = 1e-9;
        double drdt1;
        double drdtetta1;
        double dtetta2 = 1e-9;
        double drdt2;
        double drdtetta2;
        double dtob;
        double dtdt1;
        double dTEttadtetta1;

```



```

double dtdt2;
double dTEttadtetta2;
double del;
double htetta1;
double htetta2;
double Rl = 318000 + Rm;

std::vector <double> kraevaya;
while ((abs(Rl - runge_kutt_4(p, false)[18]) > 0.0001) or abs(runge_kutt_4(p,
false)[8]) > 0.00001)
{
    drob = Rl - runge_kutt_4(p, false)[18];
    p.tetta1 += dtetta1;
    drdt1 = Rl - runge_kutt_4(p, false)[18];
    p.tetta1 -= dtetta1;
    drdtetta1 = (drdt1 - drob) / dtetta1;

    p.tetta2 += dtetta2;
    drdt2 = Rl - runge_kutt_4(p, false)[18];
    p.tetta2 -= dtetta2;
    drdtetta2 = (drdt2 - drob) / dtetta2;

    dtob = runge_kutt_4(p, false)[8];
    p.tetta1 += dtetta1;
    dtdt1 = runge_kutt_4(p, false)[8];
    p.tetta1 -= dtetta1;
    dTEttadtetta1 = (dtdt1 - dtob) / dtetta1;

    p.tetta2 += dtetta2;
    dtdt2 = runge_kutt_4(p, false)[8];
    p.tetta2 -= dtetta2;
    dTEttadtetta2 = (dtdt2 - dtob) / dtetta2;
    del = abs(drdtetta1 * dTEttadtetta2 - drdtetta2 * dTEttadtetta1);

    htetta1 = (dTEttadtetta2 * drob - drdtetta2 * dtob) / del;
    htetta2 = (drdtetta1 * dtob - dTEttadtetta1 * drob) / del;
    p.tetta1 += htetta1;
    p.tetta2 += htetta2;
}
kraevaya.push_back(p.tetta1);
kraevaya.push_back(p.tetta2);
return kraevaya;
}
std::vector <double> prime(parametr p)
{
    double dt1 = 0.1;
    double dt2 = 0.1;
    double dmtdt1;
    double mtdt1;
    double mtdt;
    double dmtdt2;
    double mtdt2;
    double grad = 100;
    double a = 50;
    std::vector <double> optumys;

    while (grad > 0.00107)
    {
        std::cout << grad<<"\n";
        optumys = metod(p);
        p.tetta1 = optumys[0];
    }
}

```

```

        p.tetta2 = optumys[1];
        mtdt = runge_kutt_4(p, false)[17];
        p.t1 += dt1;
        optumys = metod(p);
        p.tetta1 = optumys[0];
        p.tetta2 = optumys[1];
        mtdt1 = runge_kutt_4(p, false)[17];
        dmtdt1 = (mtdt1 - mtdt) / dt1;

        p.t1 -= dt1;
        p.t2 += dt2;
        optumys = metod(p);
        p.tetta1 = optumys[0];
        p.tetta2 = optumys[1];
        mtdt2 = runge_kutt_4(p, false)[17];
        dmtdt2 = (mtdt2 - mtdt) / dt2;
        p.t2 -= dt2;

        grad = sqrt(dmtdt1 * dmtdt1 + dmtdt2 * dmtdt2);
        p.t1 -= a * dmtdt1;
        p.t2 -= a * dmtdt2;
        optumys = metod(p);
    }
    optumys.push_back(p.t1);
    optumys.push_back(p.t2);
    std::cout << std::setprecision(15) << p.t1 << " \n " << p.t2 << " \n" << p.tetta1 <<
"\n " << p.tetta2;
    return optumys;
}
double superprime()
{
    parametr p;
    p.tetta1 = -M_PI / 2 / (tmax - 14);
    p.tetta2 = -M_PI / 6;
    p.t1 = 450;
    p.t2 = 750;
    std::vector<double> prouz;
    std::vector<double> prouz2;
    double dM, dm, dW, dw, dm1, dm2, dw1, dw2;
    dm = 0.1;
    dw = 0.1;
    prouz = prime(p);
    p.tetta1 = prouz[0];
    p.tetta2 = prouz[1];
    p.t1 = prouz[2];
    p.t2 = prouz[3];

    dm1 = runge_kutt_4(p, false)[5] - 615;
    m_0 += dm;

    prouz2 = metod(p);
    p.tetta1 = prouz2[0];
    p.tetta2 = prouz2[1];

    dm2 = runge_kutt_4(p, false)[5] - 615 - dm;
    dM = (dm2 - dm1) / dm;

    m_0 -= dm;
    W_0 += dw;

```

```

    prouz2 = metod(p);
    p.tetta1 = prouz2[0];
    p.tetta2 = prouz2[1];

    dw2 = runge_kutt_4(p, false)[5] - 615;
    dW = (dw2 - dm1) / dw;

    std::cout << std::setprecision(15) << dM << " " << dW;

    return dM;
}
int main()
{
    setlocale(LC_ALL, "Russian");
    std::vector <double> a;
    double b;
    parametr p;
    p.tetta1 = -0.0044;
    p.tetta2 = -0.7126;
    p.t1 = 433.059;
    p.t2 = 1342.681;
    prime(p);
    //a = runge_kutt_4(p, true);
    //superprime();
    return 0;
}

```