

**Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Российский университет дружбы народов имени Патриса Лумумбы»  
Инженерная академия  
Департамент механики и процессов управления**

## ОТЧЕТ

<b>По</b>	Лабораторной работе №2 по Механике Космического Полета. Вариант 4.
-----------	---

<b>Направление:</b>	01.03.02 Прикладная математика и информатика (код направления / название направления)
<b>Профиль:</b>	Математические методы механики полета раке-носителей и космических аппаратов (название профиля)

<b>Тема:</b>	Определение возмущающего ускорения, обусловленного сопротивлением атмосферы Земли. (название лабораторной / курсовой)
--------------	--

<b>Выполнено студентом:</b>	Критским Матвеем Димитриевичем (ФИО)		
<b>Группа:</b>			ИПМбд-02-22
<b>№ студенческого:</b>			1132226149

**Москва, 2023**

### Цель работы:

Исследовать возмущения, вызываемые сопротивлением атмосферы Земли, подсчитать возмущенное ускорение и его составляющие. Увидеть зависимость возмущенного ускорения и его составляющих от уровня солнечной активности на зафиксированной высоте.

### Исходные данные:

- Информация об орбите:

Высота апоцентра( $h_a$ ) - 850 км

Высота перигея( $h_p$ ) - 350 км

Наклонение( $i$ ) - 45 град. /  $\frac{\pi}{4}$  рад. /  $\sim 0.785398$  рад.

Долгота восходящего узла( $\Omega$ ) - 20 град. /  $\frac{\pi}{9}$  рад. /  $\sim 0.349066$  рад.

Аргумент перигея( $\omega$ ) - 0 град. / 0 рад.

Средняя аномалия( $M_0$ ) - 15 град. /  $\frac{\pi}{12}$  рад. /  $\sim 0.261799$  рад.

Радиус апоцентра( $r_a$ ) -  $R_z + h_a = 7228.1$  км

Радиус перигея( $r_p$ ) -  $R_z + h_p = 6728.1$  км

Большая полуось( $a$ ) -  $\frac{r_a + r_p}{2} = 6978.1$  км

Эксцентриситет( $e$ ) -  $\frac{r_a - r_p}{r_a + r_p} = \sim 0.0358264$

Фокальный параметр( $p$ ) -  $a \cdot (1 - e^2) = 6969.14$  км

Период( $T$ ) -  $2 \cdot \pi \cdot \sqrt{\frac{a^3}{\mu}} = \sim 5801.19$  с

- Информация о планете Земля:

Радиус( $R_z$ ) - 6378.1 км

Масса( $M_z$ ) -  $5.9726 \cdot 10^{24}$  кг

Гравитационный параметр( $\mu$ ) = 398600.4415

Гравитационная постоянная( $G$ ) -  $6.674 \cdot 10^{-11} \frac{(\text{Н} \cdot \text{м}^2)}{\text{кг}^2}$

Угловая скорость( $\omega_z$ ) -  $7.2921158553 \cdot 10^{-5} \frac{\text{рад}}{\text{с}}$

Плотность ночной атмосферы на высоте 120 км( $\rho_0$ ) -  $1.58868 \cdot 10^{-8} \frac{\text{кг}}{\text{м}^3}$

Эксцентриситет Земли( $e_z^2$ ) - 0.0067385254

Большая полуось ОЗЭ( $a_z$ ) - 6378136 м

- Информация о КЛА:

Масса КЛА( $m$ ) - 1500 кг

Коэф. Силы лобового сопротивления( $C_{ха}$ ) - 2

Площадь КЛА( $S_a$ ) - 12 м<sup>2</sup>

Баллистический коэф. КЛА( $\sigma_x$ ) -  $\frac{C_{ха} \cdot S_a}{2 \cdot m} = 0.008$

### Ход работы:

1. С помощью исходных параметров орбиты найти координаты заданной точки в АГЭСК ( $x_a, y_a, z_a$ ).
2. Найти скорость и ее составляющие (трансверсальная скорость, радиальная скорость).
3. На базе значений координат АГЭСК найти положение этой точки в ГСК и рассчитать соответствующие им геодезические координаты L, В и Н.
4. Рассчитать плотность атмосферного давления  $\rho_{атм}(H)$ .
5. Найти составляющие возмущающего ускорения, обусловленного влиянием атмосферы.
6. Подвести вывод.

### Примечание:

Для расчета данных мы будем использовать 3 разные точки орбиты по времени: точка перигентра( $t = 0$  или Т(в нашем случае я взял  $t = 0$  с), точке апоцентра( $t = \frac{T}{2} \approx 2900.592775$  с) и произвольную точку орбиты ( $t = \frac{T}{13.6} \approx 426.557761$  с). В дальнейшем будем обозначать эти точки как Т<sub>р</sub>(точка перигентра), Т<sub>а</sub>(точка апоцентра), Т<sub>г</sub>(произвольная точка(“г” слова random - случайный))

**1) С помощью исходных параметров орбиты найти координаты заданной точки в АГЭСК ( $x_a, y_a, z_a$ ).**

Для расчета координат необходимо найти истинную аномалию( $\theta$ ), эксцентрическую аномалию( $E$ ), аргумент широты орбиты( $u$ ) и модуль радиус-вектора КА в АГЭСК( $r_A$ ).

Для начала рассчитаем среднюю аномалию для произвольной точки по след. формуле:  $M = M_0 + \left(\frac{2 \cdot \pi}{T}\right) \cdot t$ . Получим следующие показатели:

Для  $T_p$ :  $M \approx 0.2617993951$  рад.

Для  $T_a$ :  $M \approx 3.4033539295$  рад.

Для  $T_r$ :  $M \approx 0.7237982750$  рад.

Эксцентрическую аномалию можно найти методом приближений из уравнения Кеплера:

- 1) Задаем начальное значение  $E_0 = M$ .
- 2) Рассчитываем новое значение  $E$  по формуле:  $E_{i+1} = M + e \cdot \sin(E_i)$ , где  $i$  - номер итерации
- 3) Если  $|E_{i+1} - E_i| \leq \varepsilon$ , где  $\varepsilon$  - заранее заданное малое число (в нашем случае  $\varepsilon = 0.001$  град. /  $\frac{\pi}{180000}$  рад. /  $\sim 0.000017453$  рад.), то  $E = E_{i+1}$ , иначе  $E_i = E_{i+1}$ . Алгоритм повторяется с шага 2)

В итоге:

Для  $T_p$ :  $E \approx 0.2714034915$  рад.

Для  $T_a$ :  $E \approx 3.3943932056$  рад.

Для  $T_r$ :  $E \approx 0.7481709123$  рад.

С помощью найденных эксцентрических аномалий мы можем найти истинную аномалию и модуль радиус-вектора КА в АГЭСК по след. формулам:

$$\theta = 2 \cdot \arctan\left(\sqrt{\frac{1+e}{1-e}} \cdot \tan\left(\frac{E}{2}\right)\right)$$

$r_A = a \cdot (1 - e \cdot \cos(E))$  (или можно выразить радиус-вектор через истинную аномалию:  $\frac{a \cdot (1 - e^2)}{1 + e \cdot \cos(\theta)}$ )

Получим:

Для  $T_p$ :  $r_A \approx 6737.2509765625$  км;  $\theta \approx 0.2811797261$  рад.

Для  $T_a$ :  $r_A \approx 7220.1538085938$  км;  $\theta \approx -2.8976027966$  рад.

Для  $T_r$ :  $r_A \approx 6794.8662109375$  км;  $\theta \approx 0.7728745937$  рад.

С помощью истинной аномалии вычислим аргумент широты орбиты:  
 $u = \theta + \varpi$  (т.к.  $\varpi = 0$  согласно начальным данным, то  $u = \theta = 0.28118$ )

Теперь можно вычислить координаты АГЭСК:

$$X_a = r_A \cdot (\cos(u) \cdot \cos(\Omega) - \sin(u) \cdot \sin(\Omega) \cdot \cos(i))$$

$$Y_a = r_A \cdot (\cos(u) \cdot \sin(\Omega) + \sin(u) \cdot \cos(\Omega) \cdot \cos(i))$$

$$Z_a = r_A \cdot \sin(u) \cdot \sin(i)$$

Конечные координаты КЛА в АГЭСК:

Для  $T_p$ : [5630.1884765625; 3456.0070800781; 1321.9464111328]

Для  $T_a$ : [-6161.9443359375; -3555.2658691406; -1233.3480224609]

Для  $T_r$ : [3423.7775878906; 4816.0649414063; 3354.6198730469]

**2) Найти скорость и ее составляющие (трансверсальная скорость, радиальная скорость).**

Для упрощения выражений будем считать, что атмосфера Земли неподвижна ( $V_{\text{пер}} = 0$ ). Тогда:

$$V = V_{\text{пер}} + V_{\text{отн}} = V_{\text{отн}}$$

В проекциях на S, T, W имеет вид  $V = [-V_r, -V_\tau, 0]$ .

$V_r$  и  $V_\tau$  можно найти по след. формулам:

$$V_r = \sqrt{\frac{\mu}{p}} \cdot e \cdot \sin(\theta)$$

$$V_\tau = \sqrt{\frac{\mu}{p}} \cdot (1 + e \cdot \cos(\theta))$$

Получим:

$$\text{Для } T_p: V_r \approx 0.0751844734 \frac{\text{км}}{\text{с}}; V_\tau \approx 7.8230454710 \frac{\text{км}}{\text{с}}$$

$$\text{Для } T_a: V_r \approx -0.0654540007 \frac{\text{км}}{\text{с}}; V_\tau \approx 7.2998196607 \frac{\text{км}}{\text{с}}$$

$$\text{Для } T_r: V_r \approx 0.1891731069 \frac{\text{км}}{\text{с}}; V_\tau \approx 7.7567120028 \frac{\text{км}}{\text{с}}$$

Зная радиальную и трансверсальную скорости, можно посчитать результирующую скорость  $V$ :

$$V = \sqrt{V_r^2 + V_\tau^2} \text{ (или ее можно посчитать, не вычисляя ее составляющих по след. формуле: } \sqrt{\frac{\mu}{p}} \cdot \sqrt{1 + 2 \cdot e \cdot \cos(\theta) + e^2})$$

В результате расчетов получим след. результаты:

$$\text{Для } T_p: V \approx 7.8234067481 \frac{\text{км}}{\text{с}}$$

$$\text{Для } T_a: V \approx 7.3001131022 \frac{\text{км}}{\text{с}}$$

$$\text{Для } T_r: V \approx 7.7590184661 \frac{\text{км}}{\text{с}}$$

3) На базе значений координат АГЭСК найти положение этой точки в ГСК и рассчитать соответствующие им геодезические координаты L, В и Н.

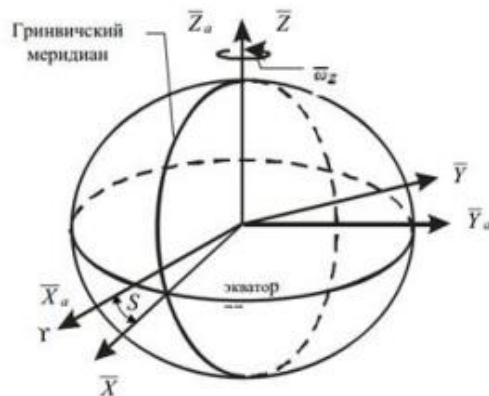


Рис. 1: Абсолютная и относительная экваториальные СК

Взаимное положение систем координат АГЭСК и ГСК определяется углом поворота Земли вокруг оси вращения  $S(t)$ , которое отвечает звездному моменту времени на момент  $t$  времени UTC. Для того, чтобы получить ГСК из АГЭСК, надо умножить координаты АГЭСК на матрицу перехода

$$S(t) = \omega_3 \cdot t$$

Для  $T_p$ :  $S(t) = 0$  рад.

Для  $T_a$ :  $S(t) \approx 0.2115120292$  рад.

Для  $T_r$ :  $S(t) \approx 0.0311050862$  рад.

$$\Psi = \begin{bmatrix} \cos(S(t)) & \sin(S(t)) & 0 \\ -\sin(S(t)) & \cos(S(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Тогда:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos(S(t)) & \sin(S(t)) & 0 \\ -\sin(S(t)) & \cos(S(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_a \\ y_a \\ z_a \end{bmatrix}$$

Или:

$$\begin{aligned} x &= \cos(S(t)) \cdot x_a + \sin(S(t)) \cdot y_a \\ y &= -\sin(S(t)) \cdot x_a + \cos(S(t)) \cdot y_a \\ z &= z_a \end{aligned}$$

Получим координаты в ГСК:

Для  $T_p$ : [5630.1884765625; 3456.0070800781; 1321.9464111328]

Для  $T_a$ : [-6771.0102539063; -2182.40625; -1233.3480224609]

Для  $T_r$ : [3571.9013671875; 4707.255859375; 3354.6198730469]

Связь координат (x, y, z) в ГСК с геодезическими (L, B, H) задаются формулами

$$x = (N + H) \cdot \cos(B) \cdot \cos(L)$$

$$y = (N + H) \cdot \cos(B) \cdot \sin(L)$$

$$z = ((1 - e_3^2) \cdot N + H) \cdot \sin(B)$$

, где

$$N = \frac{a_3}{\sqrt{1 - e_3^2 \cdot \sin^2(B)}}$$

Так как при обратном преобразовании по вышеупомянутым формулам в определенных случаях имеется деление на ноль, то необходимо использовать специальный алгоритм перехода от ГСК в геодезическую СК:

Вычисляем величину D по формуле:  $D = \sqrt{x^2 + y^2}$

1. Если  $D = 0$ , то  $B = \frac{\pi}{2} \cdot \frac{z}{|z|}$ ,  $L = 0$ ,  $H = z \cdot \sin(B) - a_3 \cdot \sqrt{1 - e_3^2 \cdot \sin^2(B)}$ . Конец алгоритма.

2. Если  $D > 0$ , то  $L_a = \arcsin\left(\frac{y}{a_3}\right)$

1) Если  $y < 0$  и  $x > 0$ , то  $L = 2 \cdot \pi - L_a$ ;

2) Если  $y < 0$  и  $x < 0$ , то  $L = \pi + L_a$ ;

3) Если  $y > 0$  и  $x < 0$ , то  $L = \pi - L_a$ ;

4) Если  $y > 0$  и  $x > 0$ , то  $L = L_a$ ;

3. Анализируем значение z:

a. Если  $z = 0$ , то  $B = 0$  и  $H = D - a_3$ . Конец алгоритма.

b. Если  $z \neq 0$ , то необходимо найти доп. величины

$$R = \sqrt{x^2 + y^2 + z^2}$$

$$C = \arcsin\left(\frac{z}{r}\right)$$

$$P = \frac{e_3^2 \cdot a_3}{2 \cdot r}$$

и реализовать итерационный процесс:

$$S_1 = 0; b = C + S_1; S_2 = \arcsin\left(\frac{P \cdot \sin(2 \cdot b)}{\sqrt{1 - e_3^2 \cdot \sin^2(b)}}\right)$$

Если  $|S_2 - S_1| < \varepsilon$ , где  $\varepsilon = 0.0001''$  (угловая секунда) /  $\frac{\pi}{648000000}$  рад - заведомо малая положительная величина, то  $B = b$ , а

$$H = D \cdot \cos(B) + z \cdot \sin(B) - a_3 \cdot \sqrt{1 - e_3^2 \cdot \sin^2(B)}$$

, иначе  $S_2 = S_1$ , и снова идет пересчет b.



Решение найдено, выход из алгоритма.

Для  $T_p$ :

$D = 6606.2856445312$  км; Т.к. координаты  $x$  и  $y$  в ГСК положительны, то  $L = L_a = 0.5505303145$ ; Т.к. координата  $z$  в ГСК не равна 0, то:

$R = 6737.2509765625$  км;  $C = 0.1974958926$  рад;  $P = 0.0031896713$ .

Тогда  $B$  считаем итерационным процессом и получаем, что  $B = 0.1987307072$ ;  $H$  считаем по формуле из пункта 3) б.. Получаем  $H = 359.947265625$ .

Для  $T_a$ :

$D = 7114.0336914063$  км; Т.к. координаты  $x$  и  $y$  в ГСК отрицательны, то  $L = \pi + L_a$  ( $L_a = -0.3118025661$ ) =  $2.8297901154$ ; Т.к. координата  $z$  в ГСК не равна 0, то:  $R = 7220.1538085938$  км;  $C = -0.1716620326$  рад.;  $P = 0.0029763377$ .

Тогда  $B$  считаем итерационным процессом и получаем, что  $B = -0.1726696640$ ;  $H$  считаем по формуле из пункта 3) б.. Получаем  $H = 842.6488037109$ .

Для  $T_r$ :

$D = 5909.0385742188$  км; Т.к. координаты  $x$  и  $y$  в ГСК положительны, то  $L = L_a = 0.9216821790$ ; Т.к. координата  $z$  в ГСК не равна 0, то:

$R = 7220.1538085938$  км;  $C = -0.1716620326$  рад.;  $P = 0.0029763377$ . Тогда  $B$  считаем итерационным процессом и получаем, что  $B = 0.5190650225$ ;  $H$  считаем по формуле из пункта 3) б.. Получаем  $H = 421.9962463379$ .

С помощью данного алгоритма получим координаты в геодезической СК:

Для  $T_p$ : [0.5505303145; 0.1987307072; 359.947265625]

Для  $T_a$ : [2.8297901154; -0.1726696640; 842.6488037109]

Для  $T_r$ : [0.9216821790; 0.5190650225; 421.9962463379]

Погрешность нахождения  $H$  по данному алгоритму не превышает 0.003 м. Положения объектов на геодезических картах представляются в геодезических координатах, кроме того, геодезические координаты применяются в теории полета космических аппаратов, в расчетах, связанных с возмущением движения КА. Использование геодезических координат и этот алгоритм позволяет решать задачи, в которых используется оценка взаимного расположения объектов на поверхности Земли и КА дистанционного зондирования Земли.

#### 4) Рассчитать плотность атмосферного давления $\rho_{\text{атм}}(H)$ .

Плотность атмосферы  $\rho$ ,  $\frac{\text{кг}}{\text{м}^3}$ , вычисляют по формуле:

$$\rho = \rho_n \cdot K_0 \cdot (1 + K_1 + K_2 + K_3 + K_4)$$

, где

$$\rho_n = \rho_0 \cdot e^{(a_0 + a_1 \cdot H + a_2 \cdot H^2 + a_3 \cdot H^3 + a_4 \cdot H^4 + a_5 \cdot H^5 + a_6 \cdot H^6)} \quad (\text{e} - \text{экспанента, а не эксцентриситет!})$$

$\rho_n$  - плотность ночной атмосферы,  $\frac{\text{кг}}{\text{м}^3}$

$a_i$  - коэф. модели, используемые для расчета плотности атмосферы при различных значениях фиксированного уровня солнечной активности  $F_0$ ;  
 $K_i$  - нормирующие коэф., учитывающие суточные, полугодовые отклонения плотности атмосферы, геомагнитную активность Солнца. Будем полагать, что  $K_0 = 1$ , а остальные  $K_i = 0$ . При таких  $K$  мы получаем, что

$$\rho = \rho_n$$

Значения  $a_i$  берутся из таблицы №1 или таблицы №2 в зависимости от  $H$ .

Таблица №1. Коэф. Модели плотности атмосферы для первого высотного диапазона (120-500 км)

Коэффициент		Значение при фиксированном уровне солнечной активности $F_0$ , $10^{-22}$ Вт/(м <sup>2</sup> · Гц)						
Обозначение	Размерность	75	100	125	150	175	200	250
$a_h$	км	120	120	120	120	120	120	120
$a_0$	-	26,8629	27,4598	28,6395	29,6418	30,1671	29,7578	30,7854
$a_1$	км <sup>-1</sup>	-0,451674	-0,463668	-0,490987	-0,514957	-0,527837	-0,517915	-0,545695
$a_2$	км <sup>-2</sup>	0,00290397	0,002974	0,00320649	0,00341926	0,00353211	0,00342699	0,00370328
$a_3$	км <sup>-3</sup>	-1,06953e-5	-1,0753e-5	-1,1681e-5	-1,25785e-5	-1,30227e-5	-1,24137e-5	-1,37072e-5
$a_4$	км <sup>-4</sup>	2,21598e-8	2,17059e-8	2,36847e-8	2,5727e-8	2,66455e-8	2,48209e-8	2,80614e-8
$a_5$	км <sup>-5</sup>	-2,42941e-11	-2,30249e-11	-2,51809e-11	-2,75874e-11	-2,85432e-11	-2,58413e-11	-3,00184e-11
$a_6$	км <sup>-6</sup>	1,09926e-14	1,00123e-14	1,09536e-14	1,21091e-14	1,25009e-14	1,09383e-14	1,31142e-14

Таблица №2. Коэф. Модели плотности атмосферы для второго высотного диапазона (500-1500 км)

Коэффициент		Значение при фиксированном уровне солнечной активности $F_0$ , $10^{-22}$ Вт/(м <sup>2</sup> · Гц)						
Обозначение	Размерность	75	100	125	150	175	200	250
$a_h$	км	500	500	500	500	500	500	500
$a_0$	-	17,8781	-2,54909	-13,9599	-23,3079	-14,7264	-4,912	-5,40952
$a_1$	км <sup>-1</sup>	-0,132025	0,0140064	0,0844951	0,135141	0,0713256	0,0108326	0,00550749
$a_2$	км <sup>-2</sup>	0,000227717	-0,00016946	-0,000328875	-0,000420802	-0,000228015	-8,10546e-5	-3,78851e-5
$a_3$	км <sup>-3</sup>	-2,2543e-7	3,27196e-7	5,05918e-7	5,73717e-7	2,8487e-7	1,15712e-7	2,4808e-8
$a_4$	км <sup>-4</sup>	1,33574e-10	-2,8763e-10	-3,92299e-10	-4,03238e-10	-1,74383e-10	-8,13296e-11	4,92183e-12
$a_5$	км <sup>-5</sup>	-4,50458e-14	1,22625e-13	1,52279e-13	1,42846e-13	5,08071e-14	3,04913e-14	-8,65011e-15
$a_6$	км <sup>-6</sup>	6,72086e-18	-2,05736e-17	-2,35576e-17	-2,01726e-17	-5,34955e-18	-4,94989e-18	1,9849e-18

В дальнейшем запись результатов расчета давления и ускорения будут записаны в виде массива, в порядке возрастания уровня солнечной

активности так, чтобы можно было сопоставить уровень солнечной активности и значения плотности и ускорения при этом уровне активности. Уровни солнечной активности = {75; 100; 125; 150; 175; 200; 250}

Для  $T_p$ :

$N = 359.947265625 \in [120; 500] \Rightarrow$  используем таблицу №1. Получим следующие значения:

$$\rho = \{1.68257 \cdot 10^{-12}; 3.00353 \cdot 10^{-12}; 4.60543 \cdot 10^{-12}; 6.47784 \cdot 10^{-12}; 8.65708 \cdot 10^{-12}; 1.10293 \cdot 10^{-11}; 1.63142 \cdot 10^{-11}\}$$

Для  $T_a$ :

$N = 842.6488037109 \in [500; 1500] \Rightarrow$  используем таблицу №2. Получим следующие значения:

$$\rho = \{1.82785 \cdot 10^{-15}; 2.6992 \cdot 10^{-15}; 4.38494 \cdot 10^{-15}; 5.98083 \cdot 10^{-15}; 9.47686 \cdot 10^{-15}; 1.4222 \cdot 10^{-14}; 2.63704 \cdot 10^{-14}\}$$

Для  $T_r$ :

$N = 421.9962463379 \in [120; 500] \Rightarrow$  используем таблицу №1. Получим следующие значения:

$$\rho = \{3.81324 \cdot 10^{-13}; 7.87354 \cdot 10^{-13}; 1.33854 \cdot 10^{-12}; 2.03294 \cdot 10^{-12}; 2.89854 \cdot 10^{-12}; 3.89201 \cdot 10^{-12}; 6.125 \cdot 10^{-12}\}$$

**5) Найти составляющие возмущающего ускорения, обусловленного влиянием атмосферы.**

Запишем модули ускорений S, T, W:

$$S = -\sigma_x \cdot \rho \cdot V \cdot V_r$$

$$T = -\sigma_x \cdot \rho \cdot V \cdot V_t$$

$$W = 0$$

Результирующее ускорение считается по формуле:

$a = \sqrt{S^2 + T^2 + W^2}$  (с этого момента “a” - это ускорение, а не большая полуось орбиты)

Ускорение силы притяжения определяется по следующему соотношению:

$$g = G \cdot \frac{M_3}{(R_3 + H)^2}$$

Подставив значения, получим 7 ускорений при 7-и разных уровнях солнечной активности(W всегда равен 0):

Для T<sub>p</sub>:

$$S = \{-7.91748 \cdot 10^{-12}; -1.41334 \cdot 10^{-11}; -2.16713 \cdot 10^{-11}; -3.04821 \cdot 10^{-11}; -4.07367 \cdot 10^{-11}; -5.18993 \cdot 10^{-11}; -7.67678 \cdot 10^{-11}\}$$

$$T = \{-8.23824 \cdot 10^{-10}; -1.4706 \cdot 10^{-9}; -2.25492 \cdot 10^{-9}; -3.1717 \cdot 10^{-9}; -4.23871 \cdot 10^{-9}; -5.4002 \cdot 10^{-9}; -7.98779 \cdot 10^{-9}\}$$

$$a = \{8.23862 \cdot 10^{-10}; 1.47066 \cdot 10^{-9}; 2.25503 \cdot 10^{-9}; 3.17185 \cdot 10^{-9}; 4.2389 \cdot 10^{-9}; 5.40044 \cdot 10^{-9}; 7.98816 \cdot 10^{-9}\}$$

Полученные графики:

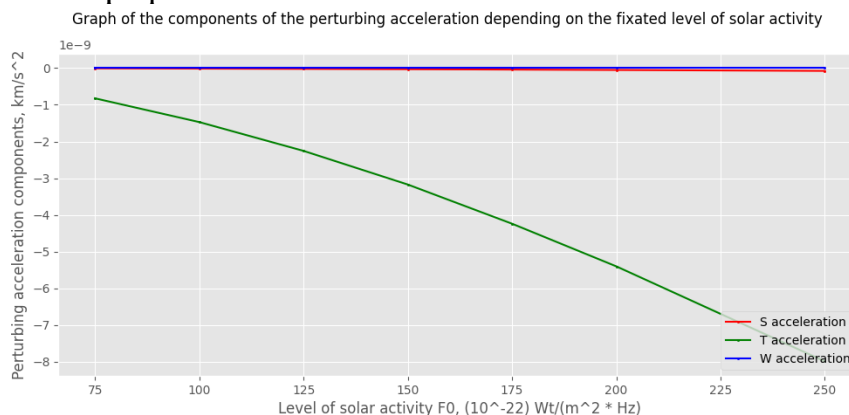


Рис.2 График составляющих возмущающего ускорения в зависимости от уровня солнечной активности(единица по Y -  $\cdot 10^{-9}$ ) в точке перигея.

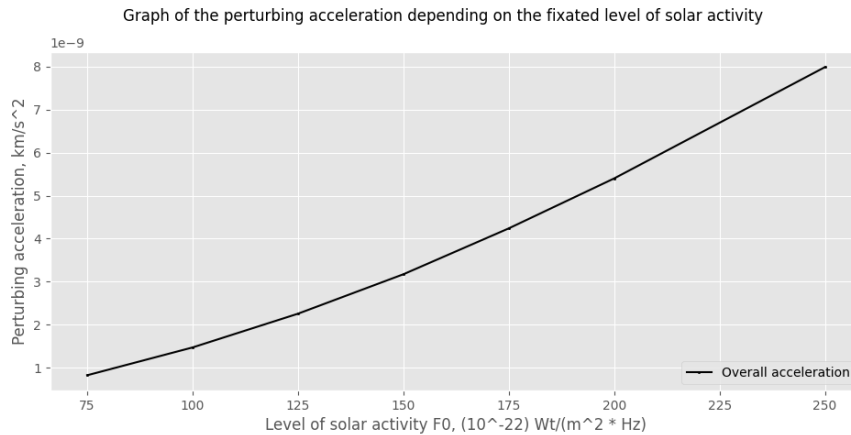


Рис.3 График возмущающего ускорения в зависимости от солнечной активности(единица по Y -  $10^{-9}$ ) в точке перигея.

Для  $T_a$ :

$$S = \{6.98807 \cdot 10^{-15}; 1.03193 \cdot 10^{-14}; 1.67641 \cdot 10^{-14}; 2.28654 \cdot 10^{-14}; 3.62311 \cdot 10^{-14}; 5.43724 \cdot 10^{-14}; 1.00817 \cdot 10^{-14}\}$$

$$T = \{-7.79241 \cdot 10^{-13}; -1.15071 \cdot 10^{-12}; -1.86937 \cdot 10^{-12}; -2.54973 \cdot 10^{-12}; -4.04014 \cdot 10^{-12}; -6.06307 \cdot 10^{-12}; -1.12422 \cdot 10^{-11}\}$$

$$a = \{7.79273 \cdot 10^{-13}; 1.15076 \cdot 10^{-12}; 1.86944 \cdot 10^{-12}; 2.54983 \cdot 10^{-12}; 4.0403 \cdot 10^{-12}; 6.06332 \cdot 10^{-12}; 1.12426 \cdot 10^{-11}\}$$

Полученные графики:

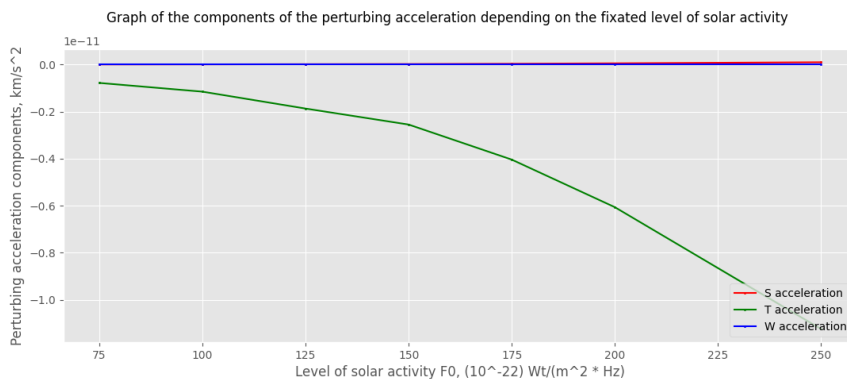


Рис.4 График составляющих возмущающего ускорения в зависимости от уровня солнечной активности(единица по Y -  $\cdot 10^{-11}$ ) в точке апоцентра.

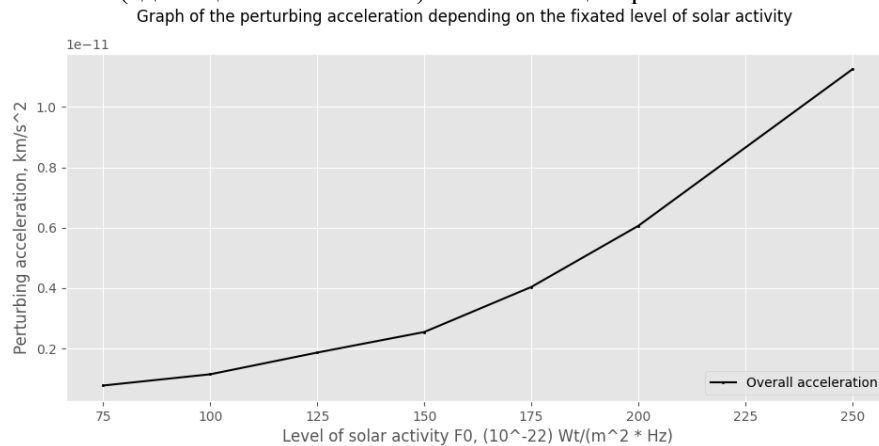


Рис.5 График возмущающего ускорения в зависимости от солнечной активности(единица по Y -  $\cdot 10^{-11}$ ) в точке апоцентра.

Для  $T_r$ :

$$S = \{-4.47765 \cdot 10^{-12}; -9.2454 \cdot 10^{-12}; -1.57176 \cdot 10^{-11}; -2.38715 \cdot 10^{-11}; \\ -3.40357 \cdot 10^{-11}; -4.57014 \cdot 10^{-11}; -7.19221 \cdot 10^{-11}\}$$

$$T = \{-1.83598 \cdot 10^{-10}; -3.79092 \cdot 10^{-10}; -6.44473 \cdot 10^{-10}; -9.78811 \cdot 10^{-10}; -1.39558 \cdot 10^{-9}; \\ -1.87391 \cdot 10^{-9}; -2.94904 \cdot 10^{-9}\}$$

$$a = \{1.83653 \cdot 10^{-10}; 3.79205 \cdot 10^{-10}; 6.44664 \cdot 10^{-10}; 9.79102 \cdot 10^{-10}; \\ 1.39599 \cdot 10^{-9}; 1.87446 \cdot 10^{-9}; 2.94992 \cdot 10^{-9}\}$$

Полученные графики:

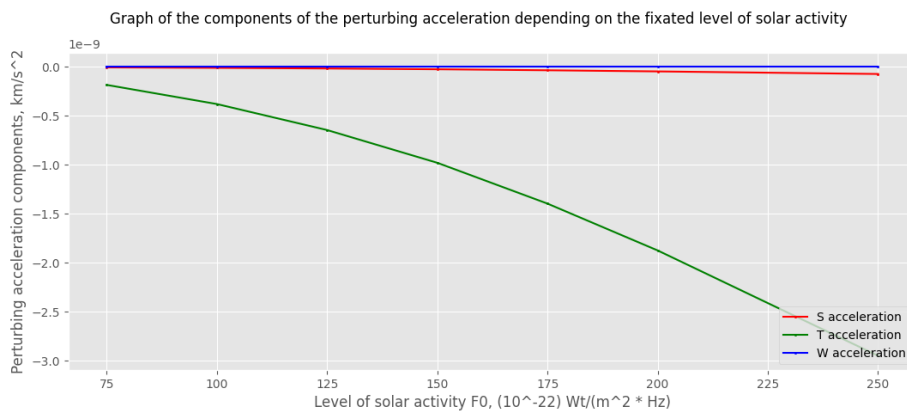


Рис.6 График составляющих возмущающего ускорения в зависимости от уровня солнечной активности(единица по Y -  $\cdot 10^{-9}$ ) в данной произвольной точке.

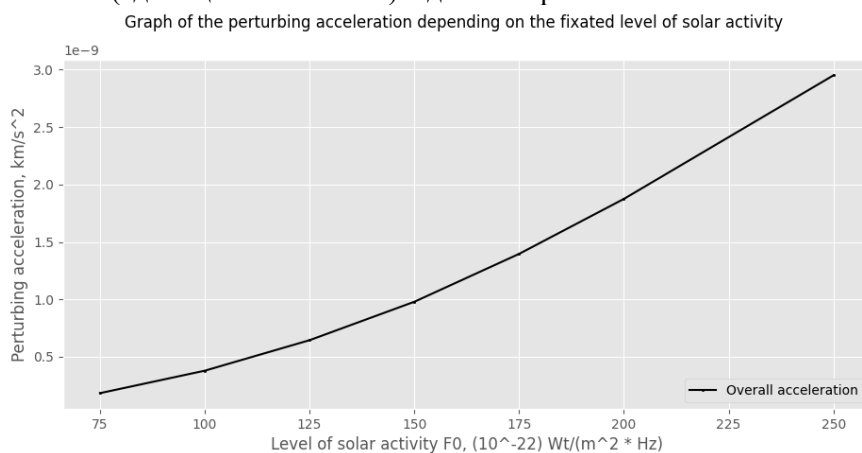


Рис.7 График возмущающего ускорения в зависимости от солнечной активности(единица по Y -  $\cdot 10^{-9}$ ) в данной произвольной точке.



### 6) Вывод.

Из пункта №4 видно, что чем выше уровень солнечной активности и чем ниже расстояние от поверхности Земли до КЛА(Н), тем больше плотность атмосферного давления и возмущающее ускорение. Т.е. в точке перицентра значения плотности атмосферы и ускорение(а также и ее составляющие) принимают максимальные значения, в точке апоцентра - минимальные, а в данной произвольной точке - значения между точкой апоцентра и перицентра(в нашем случае точка находится ближе к перицентру => значения схожи со значениями в точке перицентра).

Из пункта №2 заметно, что чем дальше КЛА от Земли, тем меньше ее скорость, принимая минимальное значение в точке апоцентра ( $V = 7.3001131022 \frac{\text{км}}{\text{с}}$ ), в перицентре - максимальное значение ( $V = 7.8234067481 \frac{\text{км}}{\text{с}}$ ) и в данной произвольной точке - значения между апоцентром и перицентром(в нашем случае  $V = 7.7590184661 \frac{\text{км}}{\text{с}}$ )  
При сравнении возмущающих ускорений с  $g$ , видно, что возмущающее ускорение сильно мало  $g$ :

```
g = 8.78013; overall perturbing acceleration = 8.23862e-10
g = 8.78013; overall perturbing acceleration = 1.47066e-09
g = 8.78013; overall perturbing acceleration = 2.25503e-09
g = 8.78013; overall perturbing acceleration = 3.17185e-09
g = 8.78013; overall perturbing acceleration = 4.2389e-09
g = 8.78013; overall perturbing acceleration = 5.40044e-09
g = 8.78013; overall perturbing acceleration = 7.98816e-09
```

Рис.8 Сравнение значения  $g$  с возмущающим ускорением при разных уровнях солнечной активности в точке перигея( $H \approx 350$  км)

```
g = 7.64548; overall perturbing acceleration = 7.79273e-13
g = 7.64548; overall perturbing acceleration = 1.15076e-12
g = 7.64548; overall perturbing acceleration = 1.86944e-12
g = 7.64548; overall perturbing acceleration = 2.54983e-12
g = 7.64548; overall perturbing acceleration = 4.0403e-12
g = 7.64548; overall perturbing acceleration = 6.06332e-12
g = 7.64548; overall perturbing acceleration = 1.12426e-11
```

Рис.9 Сравнение значения  $g$  с возмущающим ускорением при разных уровнях солнечной активности в точке апогея( $H \approx 843$  км)

```
g = 8.62063; overall perturbing acceleration = 1.83653e-10  
g = 8.62063; overall perturbing acceleration = 3.79205e-10  
g = 8.62063; overall perturbing acceleration = 6.44664e-10  
g = 8.62063; overall perturbing acceleration = 9.79102e-10  
g = 8.62063; overall perturbing acceleration = 1.39599e-09  
g = 8.62063; overall perturbing acceleration = 1.87446e-09  
g = 8.62063; overall perturbing acceleration = 2.94992e-09
```

Рис.10 Сравнение значения  $g$  с возмущающим ускорением при разных уровнях солнечной активности в данной произвольной точке( $H \approx 422$  км)

## 7) Приложение.

Ссылка на GitHub: <https://github.com/rudnmk/MSF/tree/main/2/code/lab%202>

Код программы:

Calculation.h :

```
#pragma once
#include <iostream>
#include <math.h>
#include <vector>
#include <fstream>
#include <iomanip>
#include <string>

#define PI 3.1415926

//Earth parameters
#define apogee_radius 850.0 + 6378.1 //apogee height
#define perigee_radius 350.0 + 6378.1 //perigee height
#define grav_param 398600.4415 //Gravitational parameter of the Earth
#define angular_vel 7.2921158553 * pow(10.0, -5.0) //Angular velocity of the Earth

#define i 45.0 * PI / 180.0 //inclination
#define Omega 20.0 * PI / 180.0 // ascending node longitude
#define omega 0.0 // pericenter argument
#define M0 15.0 * PI / 180.0 // average anomaly
#define accuracy 0.001 * PI / 180.0

#define SMA (((apogee_radius) + (perigee_radius)) / 2) // semi-major axis
#define ECC (((apogee_radius) - (perigee_radius)) / (2 * (SMA))) //eccentricity
#define P ((SMA) * (1 - pow((ECC), 2.0))) // focal parameter

#define night_density 1.58868 * pow(10.0, -8.0) //night density of the atmosphere on the height of 120 km
#define T 2 * PI * sqrt(pow((SMA), 3.0) / grav_param)
```

```
#define SIGMA (12.0 * 2.0) / (2.0 * 1500.0)
```

```
std::pair<double, double> calculate_anomalies(double time);  
std::pair<double, double> calculate_velocities(double THETA);
```

```
std::vector<double> calculate_AGECS(double E, double THETA);  
std::vector<double> calculate_GCS(std::vector<double> AGECS_coords, double  
time);  
std::vector<double> calculate_geodetic_coords(std::vector<double> GCS_coords,  
std::string path);
```

```
void calculate_density_and_acceleration(double H, double radial_vel, double  
transversal_vel, double vel, std::string path);
```

```
void calculation(double time);
```

Calculation.cpp:

```
#include "Calculation.h"
```

```
void calculation(double time) {  
    double E_anomaly;  
    double THETA_anomaly;  
    double radial_velocity;  
    double transversal_velocity;  
    double velocity;  
    std::string filename = "Time_" + std::to_string(time) + ".txt";  
    std::string path = "C:/Users/mk170/MSF/2/code/lab 2/lab 2 graph/" + filename;  
    std::vector<double> AGECS_coords = { 0.0, 0.0, 0.0 };  
    std::vector<double> GCS_coords = { 0.0, 0.0, 0.0 };  
    std::vector<double> geodetic_coords = { 0.0, 0.0, 0.0 };
```

```
    std::ofstream file;  
    file << std::setprecision(20);
```

```
    //anomaly calculations  
    std::pair <double, double> anomaly_calculation_result =  
    calculate_anomalies(time);  
    E_anomaly = anomaly_calculation_result.first;
```

```

    THETA_anomaly = anomaly_calculation_result.second;

    //velocities calculation
    std::pair<double, double> velocity_calculation_result =
    calculate_velocities(THETA_anomaly);
    radial_velocity = velocity_calculation_result.first;
    transversal_velocity = velocity_calculation_result.second;
    velocity = sqrt(pow(radial_velocity, 2.0) + pow(transversal_velocity, 2.0));

    //calculating spacecraft's position in different coordinate systems
    AGECS_coords = calculate_AGECS(E_anomaly, THETA_anomaly);
    GCS_coords = calculate_GCS(AGECS_coords, time);

    //calculating density and acceleration
    file.open(path);
    file << "E = " << E_anomaly << "; THETA = " << THETA_anomaly << std::endl;
    file << "Radial V = " << radial_velocity << "; Transversal V = " <<
    transversal_velocity << "; V = " << velocity << std::endl;
    file << "AGECS = [" << AGECS_coords[0] << "; " << AGECS_coords[1] << "; "
    << AGECS_coords[2] << "]" << std::endl;
    file << "GCS = [" << GCS_coords[0] << "; " << GCS_coords[1] << "; " <<
    GCS_coords[2] << "]" << std::endl;
    geodetic_coords = calculate_geodetic_coords(GCS_coords, path);
    calculate_density_and_acceleration(geodetic_coords[2], radial_velocity,
    transversal_velocity, velocity, path);
    file.close();

}

std::pair<double, double> calculate_anomalies(double time) {
    float M = M0 + (2.0 * PI / (T)) * time;
    double E_anomaly_past = M;
    double E_anomaly = M + ECC * sin(E_anomaly_past);

    while (fabs(E_anomaly - E_anomaly_past) > accuracy) {
        E_anomaly_past = E_anomaly;
        E_anomaly = M + ECC * sin(E_anomaly_past);
    }
}

```

```

double THETA_anomaly = 2 * atan(sqrt((1.0 + ECC) / (1.0 - ECC)) *
tan(E_anomaly / 2.0));

return std::make_pair(E_anomaly, THETA_anomaly);
}

std::pair<double, double> calculate_velocities(double THETA) {
double radial_velocity = sqrt(grav_param / P) * ECC * sin(THETA);
double transversal_velocity = sqrt(grav_param / P) * (1 + ECC * cos(THETA));
return std::make_pair(radial_velocity, transversal_velocity);
}

std::vector<double> calculate_AGECS(double E, double THETA) {
std::vector<double> AGECS_coords = { 0.0, 0.0, 0.0 };
double AGECS_radius_vector = SMA * (1 - ECC * cos(E));
double u = THETA + omega;

AGECS_coords[0] = AGECS_radius_vector * (cos(u) * cos(Omega) - sin(u) *
sin(Omega) * cos(i));
AGECS_coords[1] = AGECS_radius_vector * (cos(u) * sin(Omega) + sin(u) *
cos(Omega) * cos(i));
AGECS_coords[2] = AGECS_radius_vector * sin(u) * sin(i);

return AGECS_coords;
}

std::vector<double> calculate_GCS(std::vector<double> AGECS_coords, double
time) {
double ang_rotation = angular_vel * time;
std::vector<double> GCS_coords = { 0.0, 0.0, 0.0 };

GCS_coords[0] = cos(ang_rotation) * AGECS_coords[0] + sin(ang_rotation) *
AGECS_coords[1];
GCS_coords[1] = cos(ang_rotation) * AGECS_coords[1] - sin(ang_rotation) *
AGECS_coords[0];
GCS_coords[2] = AGECS_coords[2];

return GCS_coords;
}

std::vector<double> calculate_geodetic_coords(std::vector<double> GCS_coords,
std::string path) {
std::vector<double> geodetic_coords = { 0.0, 0.0, 0.0 };

```

```

std::ofstream file;
file.open(path, std::ofstream::app);
file << std::setprecision(20);
double a = 6378.136;
double e = 0.0067385254;
double x = GCS_coords[0];
double y = GCS_coords[1];
double z = GCS_coords[2];
double D = sqrt(pow(x, 2.0) + pow(y, 2.0));
if (D == 0) {
    geodetic_coords[0] = 0;
    geodetic_coords[1] = (PI / 2.0) * (z / fabs(z));
    geodetic_coords[2] = z * sin(geodetic_coords[1]) - a * sqrt(1 - e *
    pow(sin(geodetic_coords[1]), 2.0));
    file << "D = " << D << std::endl;
    file << "L = " << geodetic_coords[0] << "; B = " << geodetic_coords[1] << "; H =
    " << geodetic_coords[2] << std::endl;
    file.close();
    return geodetic_coords;
}

else {
    double L = asin(y / D);
    if (y < 0 && x > 0) {
        geodetic_coords[0] = 2 * PI - L;
    }
    else if (y < 0 && x < 0) {
        geodetic_coords[0] = PI + L;
    }
    else if (y > 0 && x < 0) {
        geodetic_coords[0] = PI - L;
    }
    else {
        geodetic_coords[0] = L;
    }

    if (z == 0) {
        geodetic_coords[1] = 0;
        geodetic_coords[2] = D - a;
        file << "D = " << D << std::endl;
        file << "L = " << geodetic_coords[0] << "; B = " << geodetic_coords[1] << "; H =
        " << geodetic_coords[2] << std::endl;
    }
}

```

```

file.close();
return geodetic_coords;
}
else {
double r = sqrt(pow(x, 2.0) + pow(y, 2.0) + pow(z, 2.0));
double c = asin(z / r);
double p = (e * a) / (2 * r);
double s1 = 0;
double b = c + s1;
double s2 = asin((p * sin(2.0 * b)) / (sqrt(1.0 - e * pow(sin(b), 2.0))));
while (fabs(s2 - s1) >= ((0.0001 / 3600.0) * PI / 180.0)){
s1 = s2;
b = c + s1;
s2 = asin((p * sin(2 * b)) / (sqrt(1 - e * pow(sin(b), 2.0))));
}
geodetic_coords[1] = b;
geodetic_coords[2] = (D * cos(b)) + (z * sin(b)) - (a * sqrt(1 - e * pow(sin(b),
2.0)));
file << "D = " << D << "; R = " << r << "; C = " << c << "; P = " << p << std::endl;
file << "L = " << geodetic_coords[0] << "; B = " << geodetic_coords[1] << "; H =
" << geodetic_coords[2] << std::endl;
file.close();
return geodetic_coords;
}
}
}

```

```

void calculate_density_and_acceleration(double H, double radial_vel, double
transversal_vel, double vel, std::string path) {
std::ofstream data_input;
std::ofstream file;
double density;
double S_acc;
double T_acc;
double W_acc = 0;
double acc;
double g = (6.67430 * 5.9726 * pow(10.0, 7.0)) / pow(6378.1 + H, 2.0);
bool lower_than_500 = 0;

```



```

double a_param_0_120[7] = { 26.8629, 27.4598, 28.6395, 29.6418, 30.1671,
29.7578, 30.7854 };
double a_param_1_120[7] = { -0.451674, -0.463668, -0.490987, -0.514957, -
0.527837, -0.517915, -0.545695 };
double a_param_2_120[7] = { 0.00290397, 0.002974, 0.00320649, 0.00341926,
0.00353211, 0.00342699, 0.00370328 };
double a_param_3_120[7] = { -1.06953 * pow(10.0, -5.0), -1.0753 * pow(10.0, -
5.0), -1.1681 * pow(10.0, -5.0), -1.25785 * pow(10.0, -5.0), -1.30227 * pow(10.0, -
5.0), -1.24137 * pow(10.0, -5.0), -1.37072 * pow(10.0, -5.0) };
double a_param_4_120[7] = { 2.21598 * pow(10.0, -8.0), 2.17059 * pow(10.0, -
8.0), 2.36847 * pow(10.0, -8.0), 2.5727 * pow(10.0, -8.0), 2.66455 * pow(10.0, -
8.0), 2.48209 * pow(10.0, -8.0), 2.80614 * pow(10.0, -8.0) };
double a_param_5_120[7] = { -2.42941 * pow(10.0, -11.0), -2.30249 * pow(10.0, -
11.0), -2.51809 * pow(10.0, -11.0), -2.75874 * pow(10.0, -11.0), -2.85432 *
pow(10.0, -11.0), -2.58413 * pow(10.0, -11.0), -3.00184 * pow(10.0, -11.0) };
double a_param_6_120[7] = { 1.09926 * pow(10.0, -14.0), 1.00123 * pow(10.0, -
14.0), 1.09536 * pow(10.0, -14.0), 1.21091 * pow(10.0, -14.0), 1.25009 *
pow(10.0, -14.0), 1.09383 * pow(10.0, -14.0), 1.31142 * pow(10.0, -14.0) };

```

```

double a_param_0_500[7] = { 17.8481, -2.54909, -13.9599, -23.3079, -14.7264, -
4.912, -5.40952 };
double a_param_1_500[7] = { -0.132025, 0.0140064, 0.0844951, 0.135141,
0.0713256, 0.0108326, 0.00550749 };
double a_param_2_500[7] = { 0.000227717, -0.00016946, -0.000328875, -
0.000420802, -0.000228015, -8.10546 * pow(10.0, -5.0), -3.78851 * pow(10.0, -
5.0) };
double a_param_3_500[7] = { -2.2543 * pow(10.0, -7.0), 3.27196 * pow(10.0, -
7.0), 5.05918 * pow(10.0, -7.0), 5.73717 * pow(10.0, -7.0), 2.8487 * pow(10.0, -
7.0), 1.15712 * pow(10.0, -7.0), 2.4808 * pow(10.0, -8.0) };
double a_param_4_500[7] = { 1.33574 * pow(10.0, -10.0), -2.8763 * pow(10.0, -
10.0), -3.92299 * pow(10.0, -10.0), -4.03238 * pow(10.0, -10.0), -1.74383 *
pow(10.0, -10.0), -8.13296 * pow(10.0, -11.0), 4.92183 * pow(10.0, -12.0) };
double a_param_5_500[7] = { -4.50458 * pow(10.0, -14.0), 1.22625 * pow(10.0, -
13.0), 1.52279 * pow(10.0, -13.0), 1.42846 * pow(10.0, -13.0), 5.08071 *
pow(10.0, -14.0), 3.04913 * pow(10.0, -14.0), -8.65011 * pow(10.0, -15.0) };
double a_param_6_500[7] = { 6.72086 * pow(10.0, -18.0), -2.05736 * pow(10.0, -
17.0), -2.35576 * pow(10.0, -17.0), -2.01726 * pow(10.0, -17.0), -5.34955 *
pow(10.0, -18.0), -4.94989 * pow(10.0, -18.0), 1.9849 * pow(10.0, -18.0) };

```

```

if (H < 500) {
lower_than_500 = 1;

```

```

}

data_input.open("C:/Users/mk170/MSF/2/code/lab 2/lab 2
graph/acceleration_data.txt");
file.open(path, std::ofstream::app);
for (int step = 0; step < 7; step++) {
if (lower_than_500) {
density = night_density * exp(a_param_0_120[step] + a_param_1_120[step] * H +
a_param_2_120[step] * pow(H, 2.0) + a_param_3_120[step] * pow(H, 3.0) +
a_param_4_120[step] * pow(H, 4.0) + a_param_5_120[step] * pow(H, 5.0) +
a_param_6_120[step] * pow(H, 6.0));
}
else {
density = night_density * exp(a_param_0_500[step] + a_param_1_500[step] * H +
a_param_2_500[step] * pow(H, 2.0) + a_param_3_500[step] * pow(H, 3.0) +
a_param_4_500[step] * pow(H, 4.0) + a_param_5_500[step] * pow(H, 5.0) +
a_param_6_500[step] * pow(H, 6.0));
}
S_acc = (-1.0) * SIGMA * density * vel * radial_vel * 1000.0;
T_acc = (-1.0) * SIGMA * density * vel * transversal_vel * 1000.0;
acc = sqrt(pow(S_acc, 2.0) + pow(T_acc, 2.0));
data_input << S_acc << " " << T_acc << " " << W_acc << " " << acc << std::endl;
file << "Density = " << density << "; Vector a = {" << S_acc << "; " << T_acc <<
"; " << W_acc << "}; a = " << acc << std::endl;
std::cout << " g = " << g << "; overall perturbing acceleration = " << acc <<
std::endl;
}
data_input.close();
file.close();
}

```

lab.cpp

```
#include "Calculation.h"
```

```

int main() {
//можно менять, что именно подается в функцию, убирая "/" перед вызовом
функции или меняя значения.
double time = T / 13.6;
double perigee_time = 0;
double apogee_time = T / 2.0;
//calculation(perigee_time);

```

```
//calculation(time);  
//calculation(apogee_time);  
return 0;  
}
```