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

ОТЧЕТ

По **Механика космического полета. Лабораторная работа №1. Вариант 3**

Направление: **01.03.02 Прикладная математика и информатика**
(код направления / название направления)

Профиль: **Математические методы механики полета ракет-носителей и космических аппаратов**
(название профиля)

Тема: **Уравнение Кеплера**
(название лабораторной / курсовой)

Выполнено студентом: **Жарова Полина Кирилловна**
(ФИО)

Группа: **ИПМбд-01-22**
№ студенческого: 1132226165

Москва, 2023

Интересные факты о миссии МКС.

1. МКС является самым дорогим проектом за всю историю человечества. В его строительство и содержание вложено около 150000000000$.
2. МКС планируют закрыть в 2031 году. Это обусловлено тем, что многие модули МКС уже устарели, т.к. технологии шагнули далеко вперед с момента их создания.

Таблица с данными, необходимым для написания программы.

№ вар.	Миссия	$r_п$, км	r_a, км	a, км	e	τ, с	Планета	$M_{пл}$, * 10^{24} кг
3	МКС	6786	6793	6789.5	0.0005155	0	Земля	5.97

Из открытых источников находим высоту перигея и апогея орбиты МКС = 415 и 422 км соответственно.

Прибавляя $R_{Земли} \approx 6371$ км, получаем r_a и $r_п$ (в таблице).

a (большую полуось орбиты) находим по формуле $a = (r_a + r_п)/2$.

Эксцентриситет орбиты e находим по формуле $e = (r_a - r_p)/(r_a + r_p)$.

Также нам необходимо найти среднюю угловую скорость: $n = \sqrt{\frac{MG}{a^3}}$, где M – масса планеты (Земли), G – гравитационная постоянная, a – большая полуось орбиты.

И период обращения КА по орбите вокруг центрального небесного тела: $T = 2\pi * \sqrt{\frac{a^3}{MG}} = 2\pi/n$.

Следовательно, T = 5567.6 с, n = 0.00112853 рад/с (высчитывается в программе).

Код программы см. в Приложения.

Графики истинной (θ), эксцентрической (E) и средней (M) аномалий, полученные методом итераций.

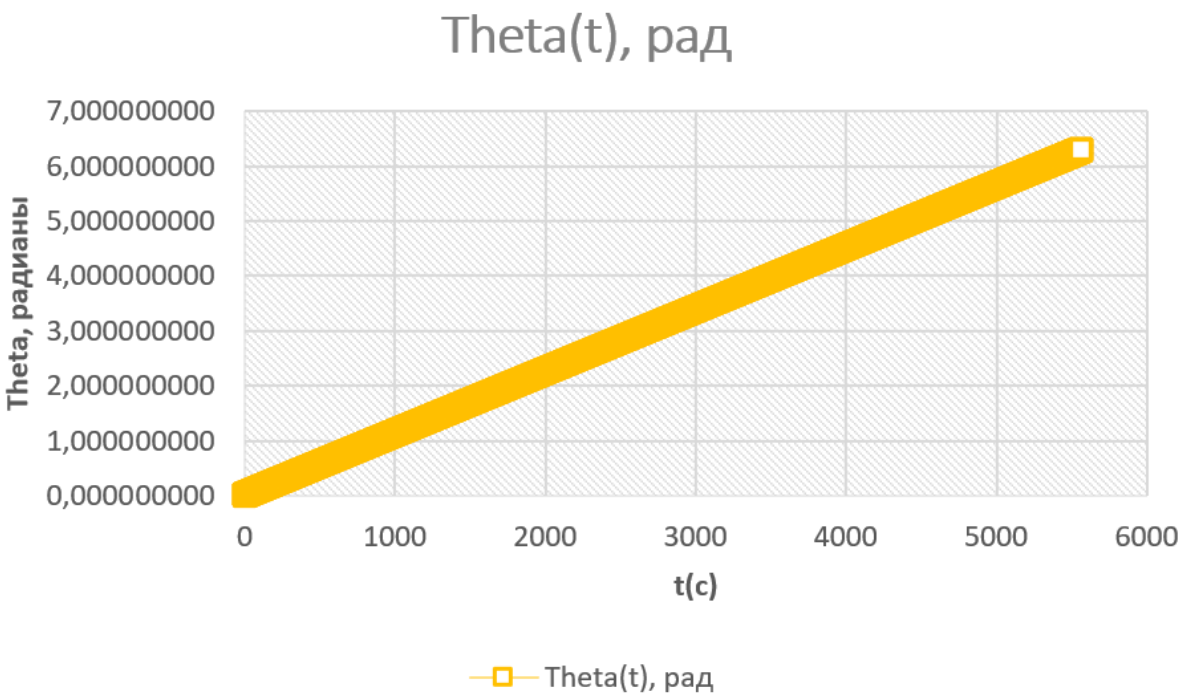

График зависимости истинной аномалии (θ, рад.) от времени (t, с).

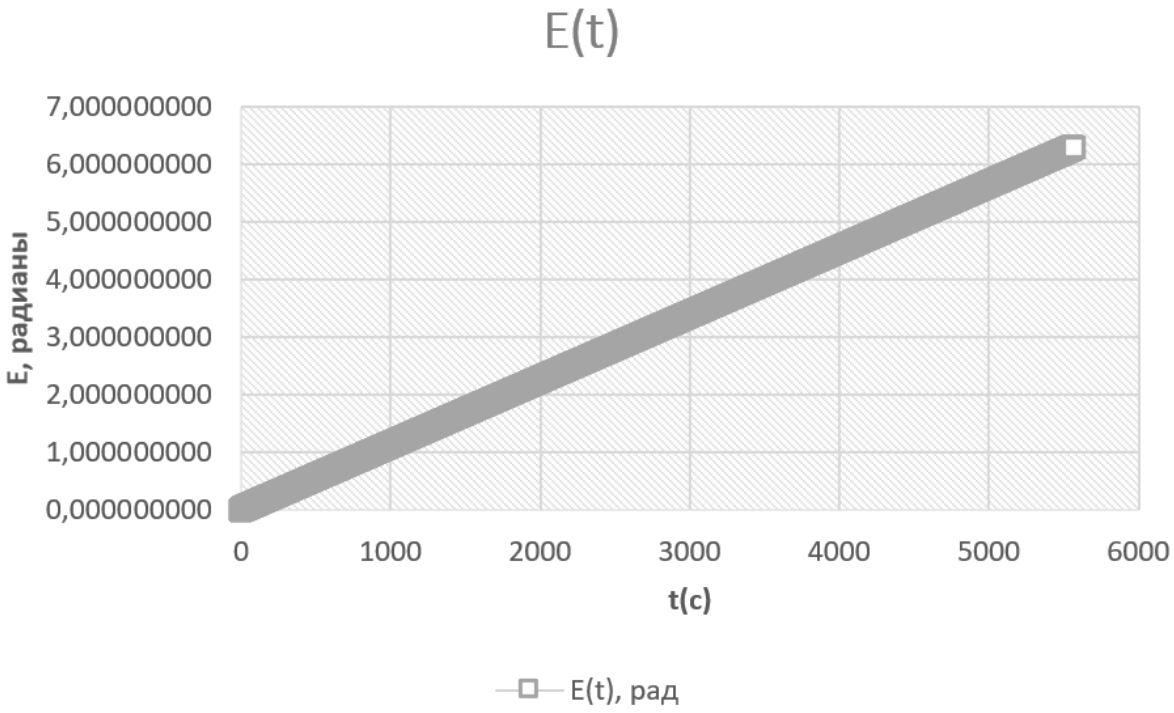

График зависимости эксцентрической аномалии (E, рад.) от времени (t, c).

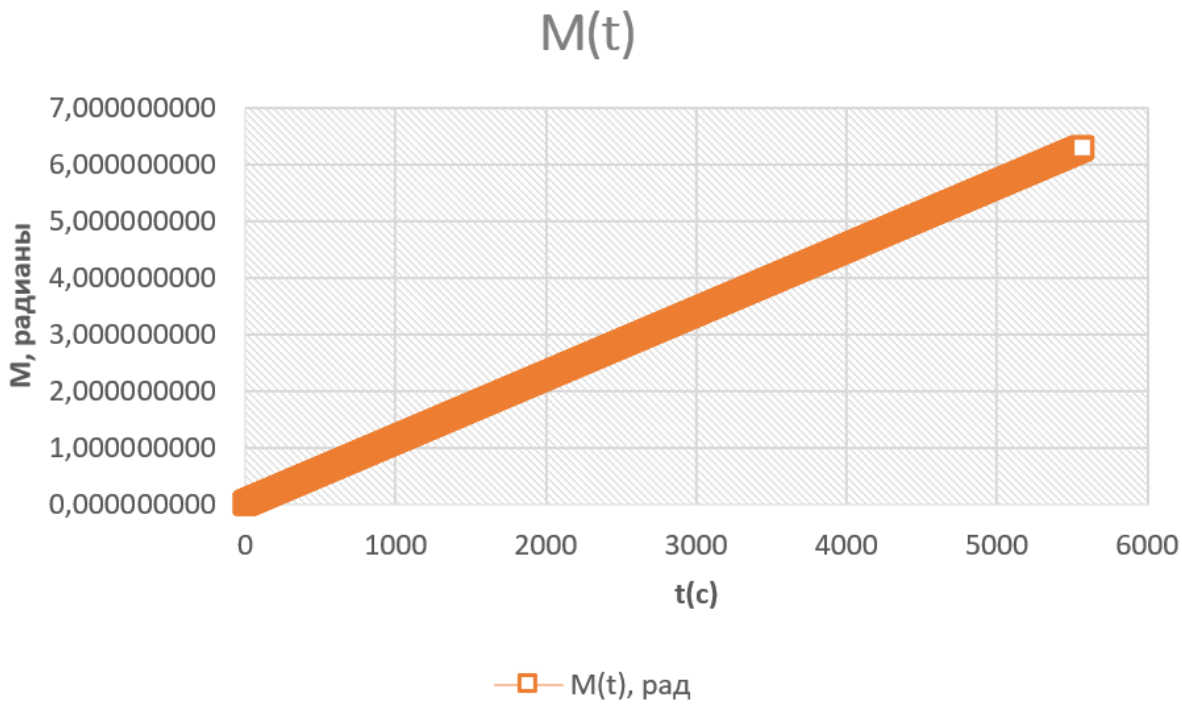

График зависимости средней аномалии (M, рад.) от времени (t, c).

Графики истинной (θ), эксцентрической (E) и средней (M) аномалий, полученные методом деления отрезка пополам.

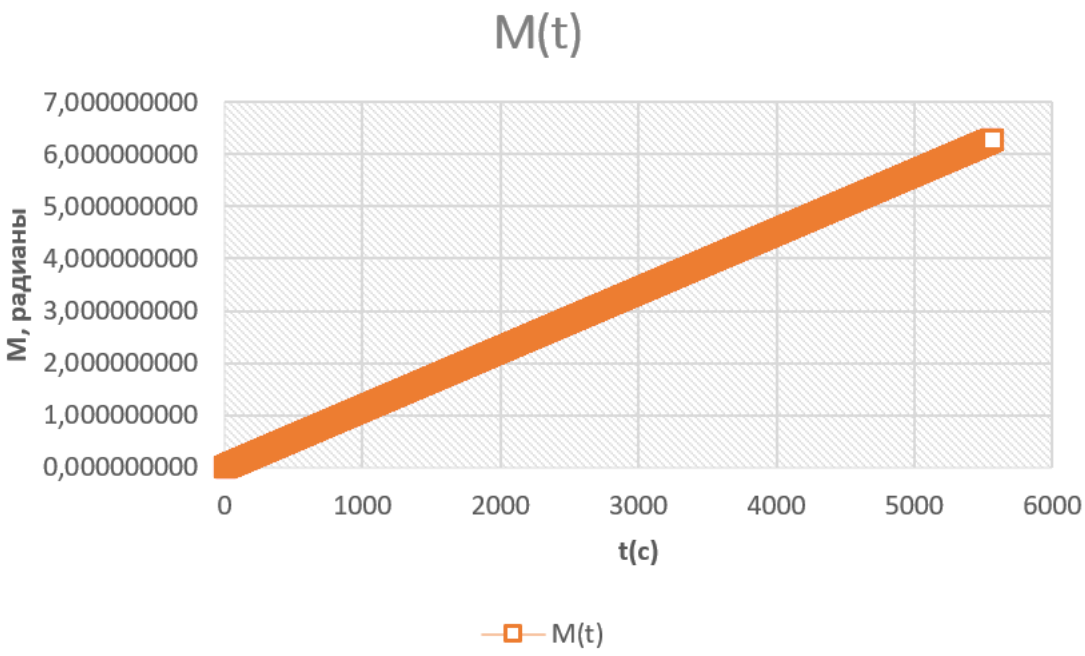

График зависимости средней аномалии (M, рад.) от времени (t, c).

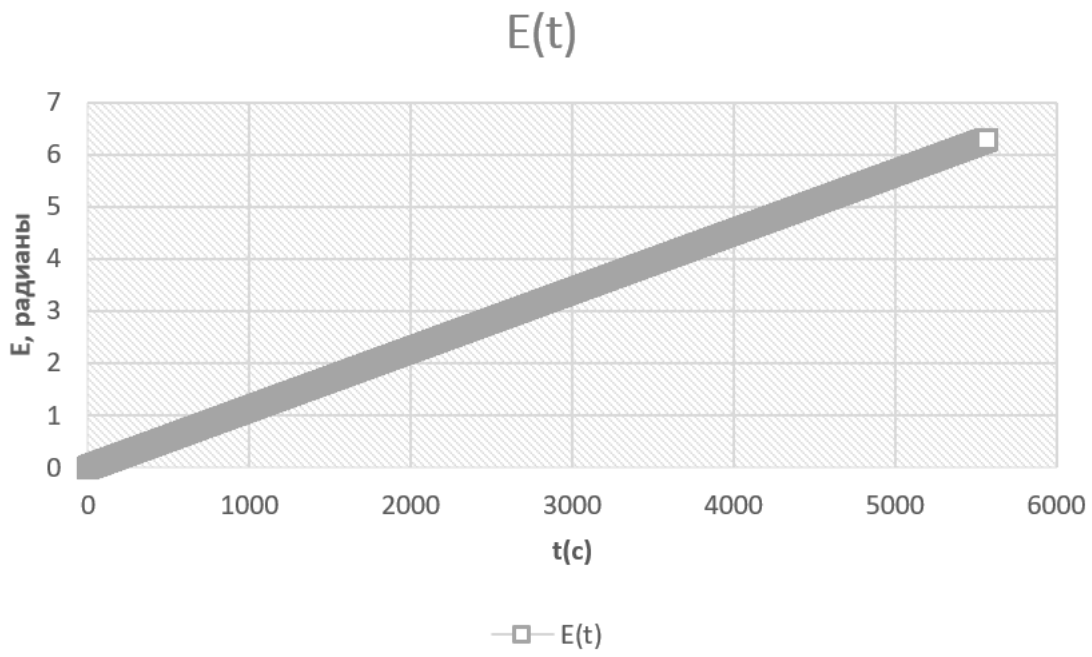

График зависимости эксцентрической аномалии (E, рад.) от времени (t, c).

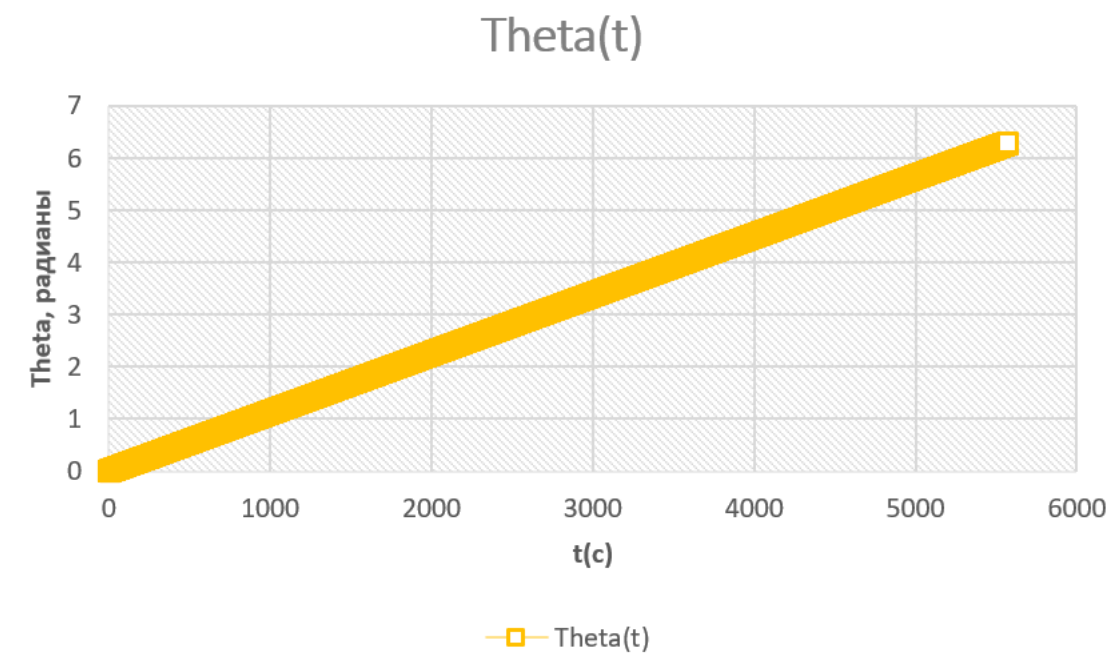

График зависимости истинной аномалии (θ, рад.) от времени (t, с).

Графики истинной (θ), эксцентрической (E) и средней (M) аномалий, полученные методом деления отрезка в соотношение золотого сечения.

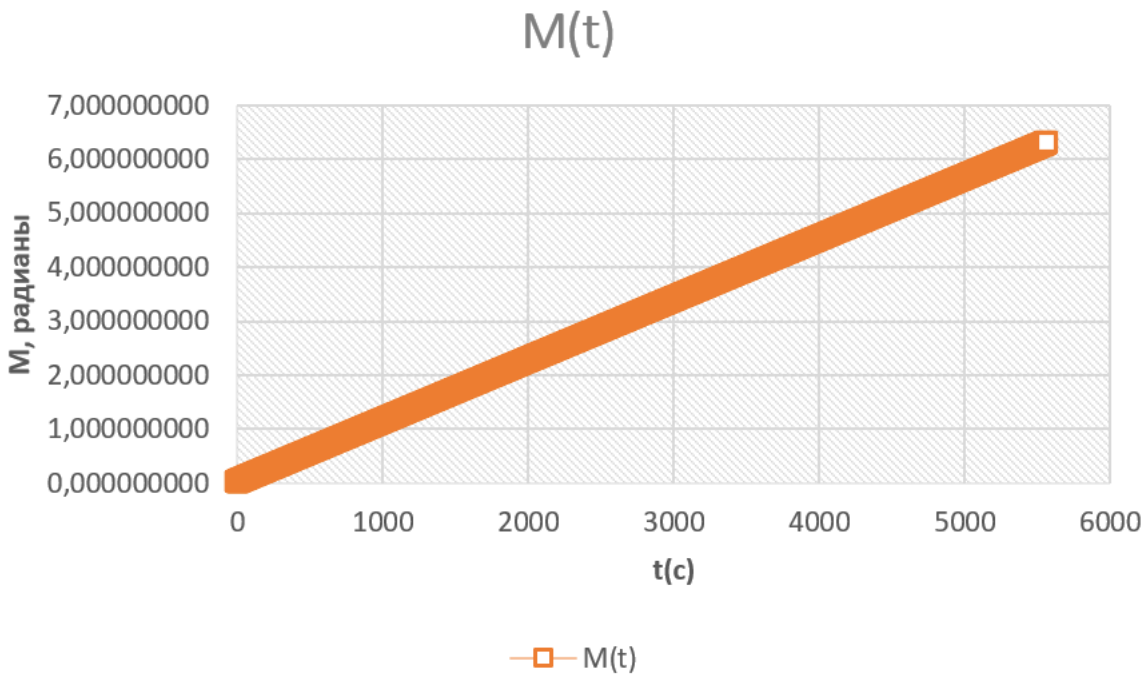

График зависимости средней аномалии (M, рад.) от времени (t, с).

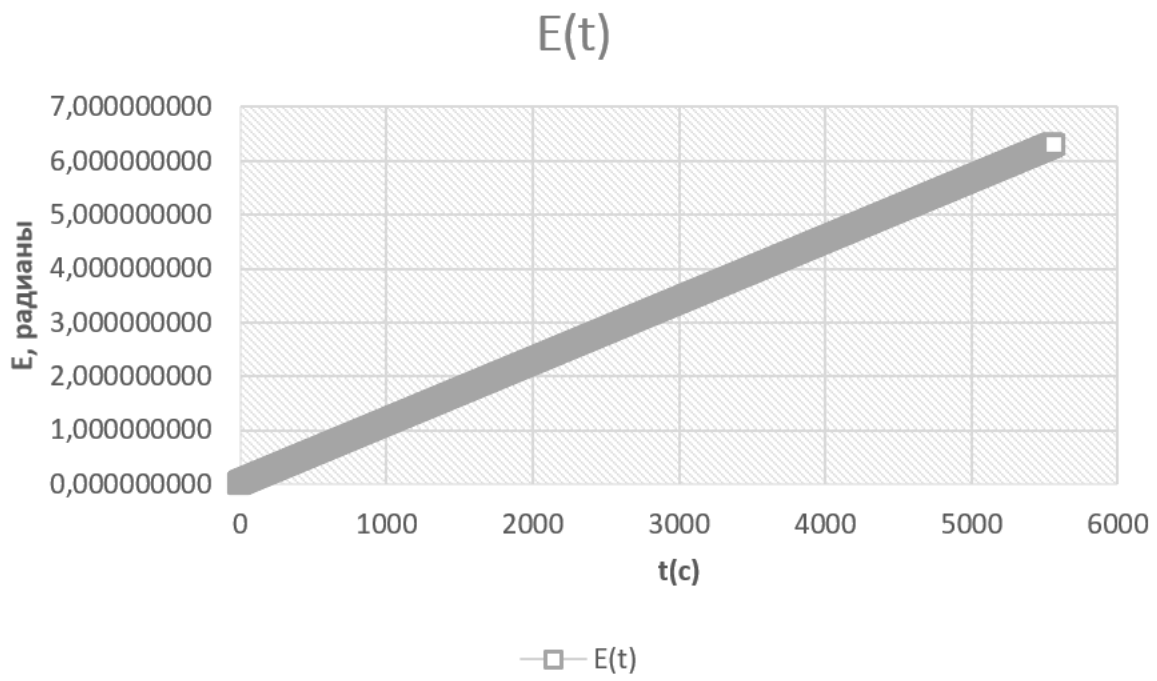

График зависимости эксцентрической аномалии (E, рад.) от времени (t, с).

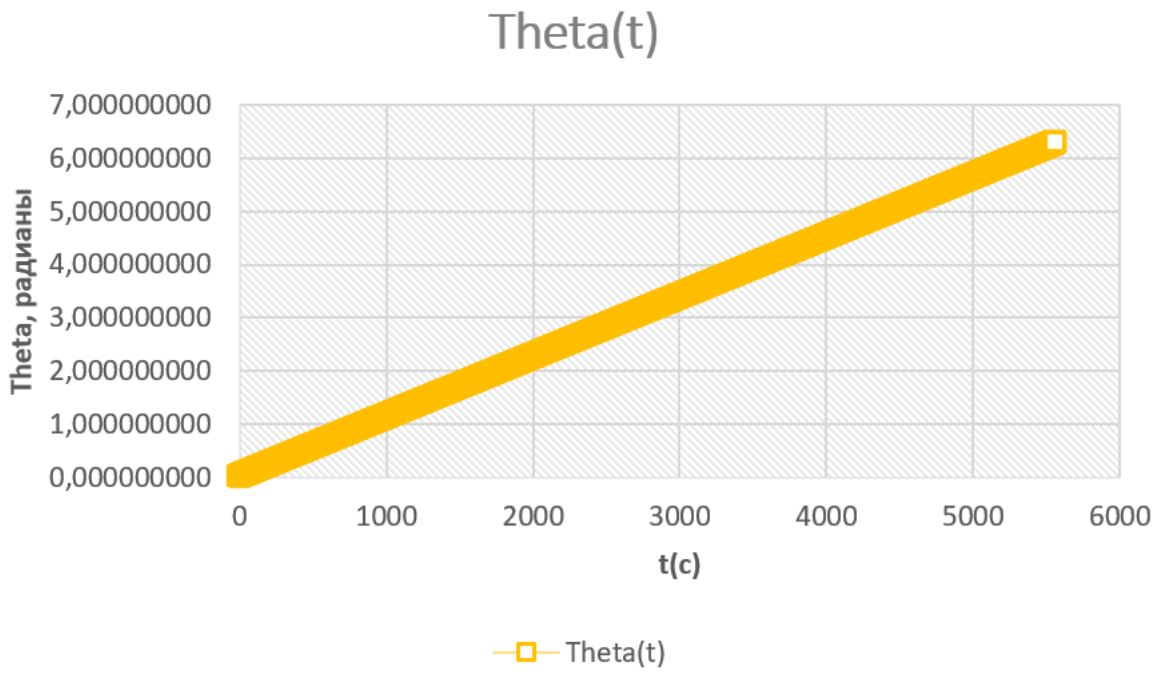

График зависимости истинной аномалии (θ, рад.) от времени (t, с).

Графики истинной (θ), эксцентрической (E) и средней (M) аномалий, полученные методом Ньютона.

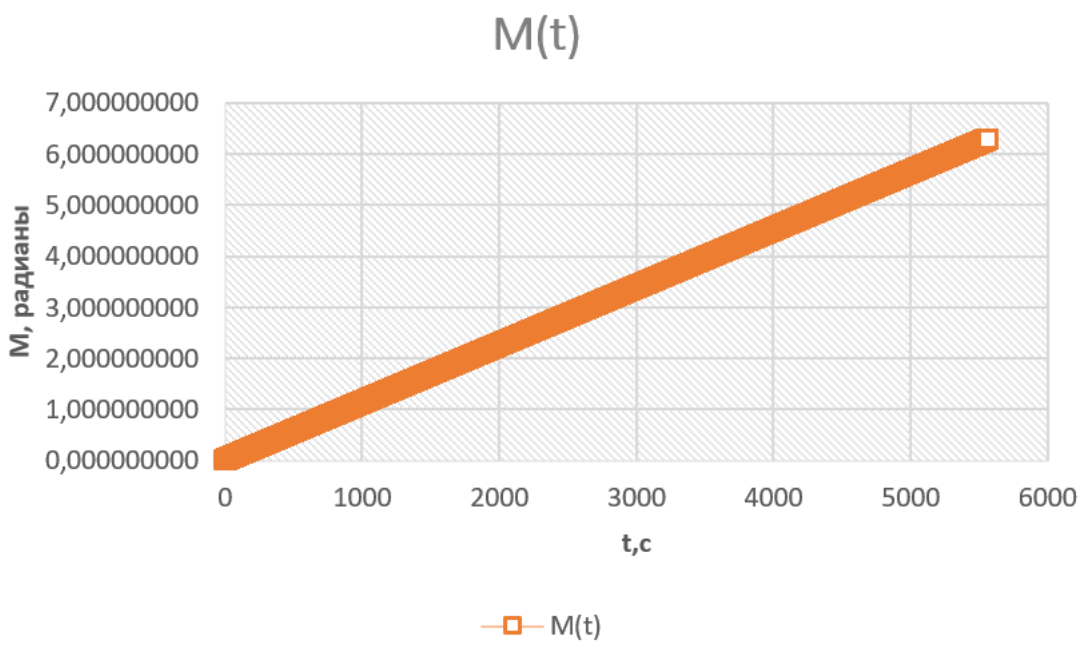

График зависимости средней аномалии (M, рад.) от времени (t, с).

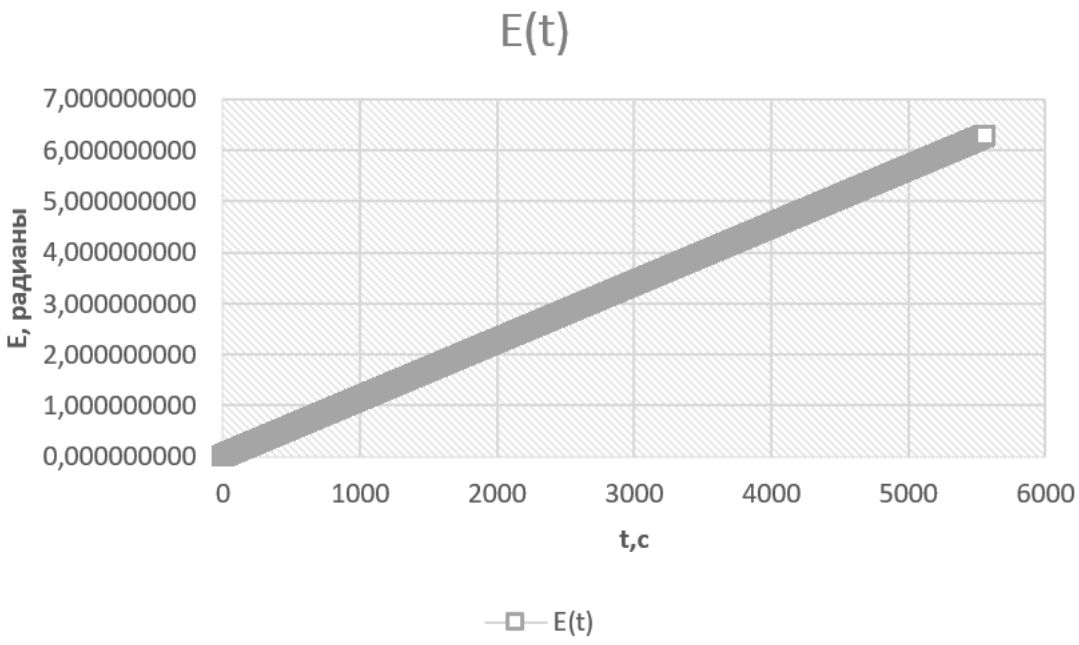

График зависимости эксцентрической аномалии (E, рад.) от времени (t, с).

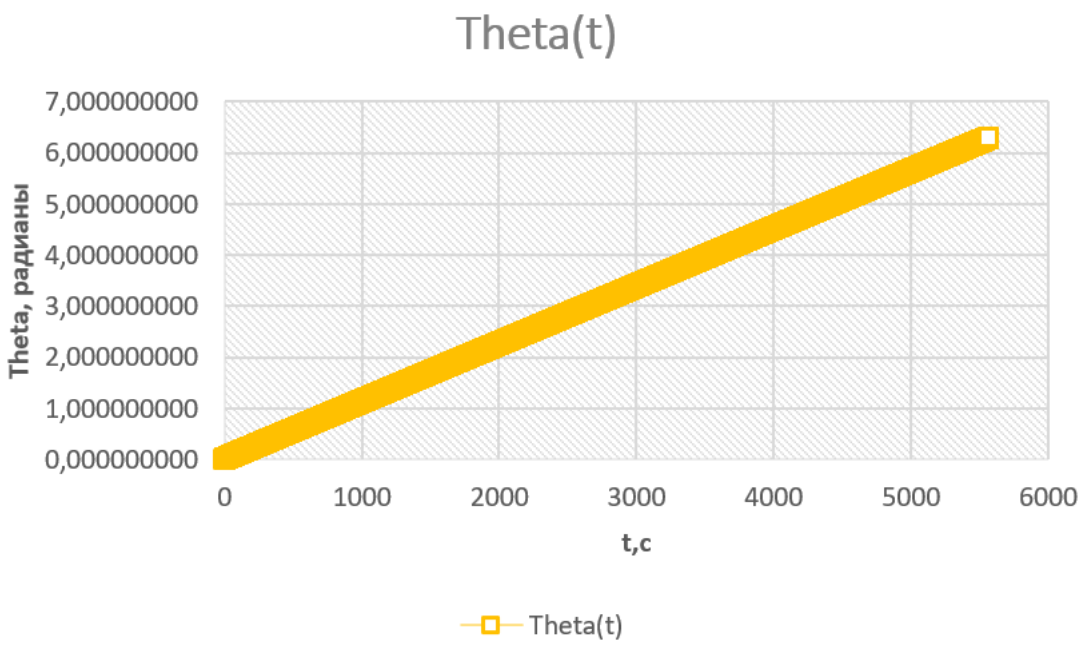

График зависимости истинной аномалии (θ, рад.) от времени (t, с).

Вывод: полученные графики представляют собой монотонно возрастающую прямую *(т.к. орбита МКС практически круговая, перегиб, который должен быть на графиках E и θ при прохождении апогея за счет отрицательности синуса, входящего в формулу для E (эксцентрической аномалии) и θ (истинной аномалии), отсутствует из-за малой величины e (эксцентриситета), первая значащая цифра которого находится в десятитысячной части десятичной записи)*, которая имеет минимум (= 0) в точке начала отсчета (перигей, t = 0 с), в апогее = π, а при повторном прохождении перигея (совершив один полный оборот) прямая достигает значения 2π (точка максимума – t = 5567 с), что является для графиков максимумом.

Приложения

```
#include <iostream>
#include <cmath>
#include <fstream>
#include <iomanip>

#define EPS 0.00001
#define PI 3.14159265359

double iterations(double e, double M);
double popolam(double e, double M);
double golden(double e, double M);
double newton(double e, double M);

int main()
{
	//parametri orbiti
	double Ra, Rp, a, e, GM, n, T, M, E, true_anomaly, p, Vp, Vr, V, r;
	Ra = 6793;
	Rp = 6786;
	a = (Ra + Rp) / 2;
	e = (Ra - Rp) / (Ra + Rp);
	GM = 398600.4415;
	n = sqrt(GM / pow(a, 3));
	T = 2 * PI / n;
	p = a * (1 - pow(e, 2));


	//file creation
	std::ofstream fout;

	try {
		fout.open("Data_newton.txt");
	}
	catch (std::exception& ex) {
		std::cout << "File opening error.";
		return 0;
	}

	//vivod v file poluchennih znacheniy anomaliy
	fout << "t, c\t";
	fout << "M(t), рад\t";
	fout << "E(t), рад\t";
	fout << "Theta(t), рад\t";
	fout << "Vp, (км/с)\t";
	fout << "Vr, (км/с)\t";
	fout << "V, (км\с)\t";
	fout << "r (км)" << std::endl;

	for (int t{ 0 }; t <= T; t++) {
		fout << t << "\t";
		M = n * t;
		E = newton(e, M);
		true_anomaly = atan(sqrt((1 + e) / (1 - e)) * tan(E / 2)) * 2;

		if (true_anomaly < 0)
			true_anomaly += 2 * PI;

		/*double cosinus = cos(true_anomaly);
		if (cosinus < 0)
			cosinus += 2 * PI;

		double sinus = sin(true_anomaly);
		if (sinus < 0)
```

```
                        sinus += 2 * PI;*/

                Vp = sqrt(GM/p) * (1 + (e * cos(true_anomaly)));
                Vr = (sqrt(GM / p)) * e * sin(true_anomaly);
                V = sqrt(pow(Vr, 2) + pow(Vp, 2));
                r = p / (1 + e * cos(true_anomaly));

                fout << M << "\t";
                fout << E << "\t";
                fout << true_anomaly << "\t";
                fout << Vp << "\t";
                fout << Vr << "\t";
                fout << V << "\t";
                fout << r << std::endl;

        }

        fout.close();
        std::cout << "The data was successfully written down in the 'Data.txt'
file.";

        return 0;
}


//E metodom iteraciy
double iterations(double e, double M) {

        double Ek1, Ek = M;

        for (int i{ 0 }; i < 50; i++) {

                Ek1 = e * sin(Ek) + M;

                if (abs(Ek1 - Ek) < EPS)
                        return Ek1;

                Ek = Ek1;
        }

        return 0;
}


//E metodom polovinnogo deleniya
double popolam(double e, double M) {

        double a = M - 2, b = M + 2, c;

        if (((a - e * sin(a) - M) * (b - e * sin(b) - M)) < 0) {

                for (int i{ 0 }; i < 50; i++) {

                        c = (a + b) / 2;

                        if (abs(c - e * sin(c) - M) < EPS)
                                return c;

                        if ((a - e * sin(a) - M) * (c - e * sin(c) - M) < 0)
                                b = c;
                        else
                                a = c;
                }
        }
        return 0;
}
```

```
//E metodom deleniya v sootnoshenii zolotogo secheniya
double golden(double e, double M) {

	double a = M - 3, b = M + 3, c;

	if (((a - e * sin(a) - M) * (b - e * sin(b) - M)) < 0) {

		for (int i{ 0 }; i < 100; i++) {

			c = a + ((b - a) / ((sqrt(5) + 1) / 2));

			if (abs(c - e * sin(c) - M) < EPS)
				return c;

			if ((a - e * sin(a) - M) * (c - e * sin(c) - M) < 0)
				b = c;
			else
				a = c;
		}
	}
	return 0;
}

//E metodom Newtona
//proizvodnuyu vzyla ne sovsem korrektno poluchaetsya (ne Ek, a E(k-1), no
znacheniya OK. it works)
double newton(double e, double M) {

	double Ek1, Ek = M, f, f1, temp_Ek1, temp_Ek;

	for (int i{ 0 }; i < 50; i++) {
		if (i == 0)
			Ek1 = Ek - ((Ek - e * sin(Ek) - M) / (1 - e * cos(Ek)));
		else
			Ek1 = Ek - ((Ek - e * sin(Ek) - M) / ((f1 - f) / (temp_Ek1 -
temp_Ek)));

		if (abs(Ek1 - Ek) < EPS)
			return Ek1;

		f = Ek - e * sin(Ek) - M;
		f1 = Ek1 - e * sin(Ek1) - M;
		temp_Ek = Ek;
		temp_Ek1 = Ek1;

		Ek = Ek1;
	}
	return 0;
}
```