**Федеральное государственное автономное образовательное учреждение**

**высшего образования**

**«Российский университет дружбы народов имени Патриса Лумумбы»**

Инженерная академия

*Департамент механики и процессов управления*

**ОТЧЕТ**

|  |  |
| --- | --- |
| **По** | **Лабораторной работе** |

|  |  |
| --- | --- |
| **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:** | **Ляшенко Амалия Романовна** | | |
|  | (ФИО) | | |
| **Группа:** |  | **ИПМбд-01-22** |  |
| **№ студенческого:** | | 1132226167 |  |

**Москва, 2023**

**Содержание**

[Теоретическая часть для отчета 2](#_Toc152921159)

[Вводные данные: 2](#_Toc152921160)

[Расчёты: 3](#_Toc152921161)

[Интересные факты: 3](#_Toc152921162)

[Факт 1: 3](#_Toc152921163)

[Факт 2: 3](#_Toc152921164)

[Соответствие переменных задачи и идентификаторам 3](#_Toc152921165)

[Дополнение 5](#_Toc152921166)

[Вывод 9](#_Toc152921167)

[Приложение 9](#_Toc152921168)

# Теоретическая часть для отчета

Миссия: Луна-10

## Вводные данные:

ra = 1017 км.

rp = 350 км.

## Расчёты:

*–* большая полуось орбиты

*-*эксцентриситет орбиты

## Интересные факты:

Факт 1: Миссия Луна-10, запущенная Советским Союзом в 1966 году, стала первым успешным космическим аппаратом, который достиг поверхности Луны. Это открыло двери для дальнейших исследований Луны и помогло нам расширить наши познания о космической механике и навигации.

Факт 2: Во время миссии Луна-10 была использована уникальная техника маневрирования, известная как "лунное торможение". Это позволило космическому аппарату изменить свою орбиту, используя гравитацию Луны, чтобы затормозить и успешно спуститься на поверхность, открывая новые методы и подходы в механике космического полета.

# Соответствие переменных задачи и идентификаторам

|  |  |  |
| --- | --- | --- |
| **Наименование переменной** | **Обозначение в документе** | **Обозначение в программе** |
| гравитационный параметр Луны, / |  |  |
| большая полуось орбиты, км |  |  |
| эксцентриситет орбиты |  |  |
| Бесконечно малая | 𝜀 | epsilon |
| фокальный параметр орбиты, км |  |  |
| Средняя угловая скорость, км/с | *n* | *n* |
| период движения по орбите, c |  |  |
| текущее время, с |  |  |
| Эксцентрическая аномалия, рад |  |  |
| угол истинной аномалии, рад |  |  |
| Истинная аномалия, рад | 𝜗 | true\_anomaly |
| Средняя аномалия, рад | *M* | *M* |
| текущий радиус-вектор, км |  |  |
| Радиус апогея | a | *r\_a* |
| Радиус перицентра | p | *r\_p* |
| трансверсальная скорость, км/с |  |  |
| радиальная скорость, км/с |  |  |
| модуль скорости, км/с |  |  |

## Дополнение

M(t) – зависимость средней аномалии (M) от времени прохождения (t) полного оборота от апоцентра

E(t) – зависимость эксцентрической аномалии (E) от времени прохождения (t) полного оборота от апоцентра

- зависимость истинной аномалии () от времени прохождения (t) полного оборота от апоцентра

V(t) – вектор скорости спутника

Vr(t)– радиальная компонента вектора скорости спутника (направлена вдоль радиус-вектора r)

Vn(t) - поперечная (трансверсальную) компонента вектора скорости спутника (направлена перпендикулярно радиус-вектору r)

**Графическое представление результатов**

Ниже представлены графики зависимостей искомых величин для 40 последовательных моментов t (40 итераций).

Кроме того, изображен график зависимости скорости и ее компонент на одном рисунке для наглядности разницы величин и график зависимости эксцентрической аномалии от времени.

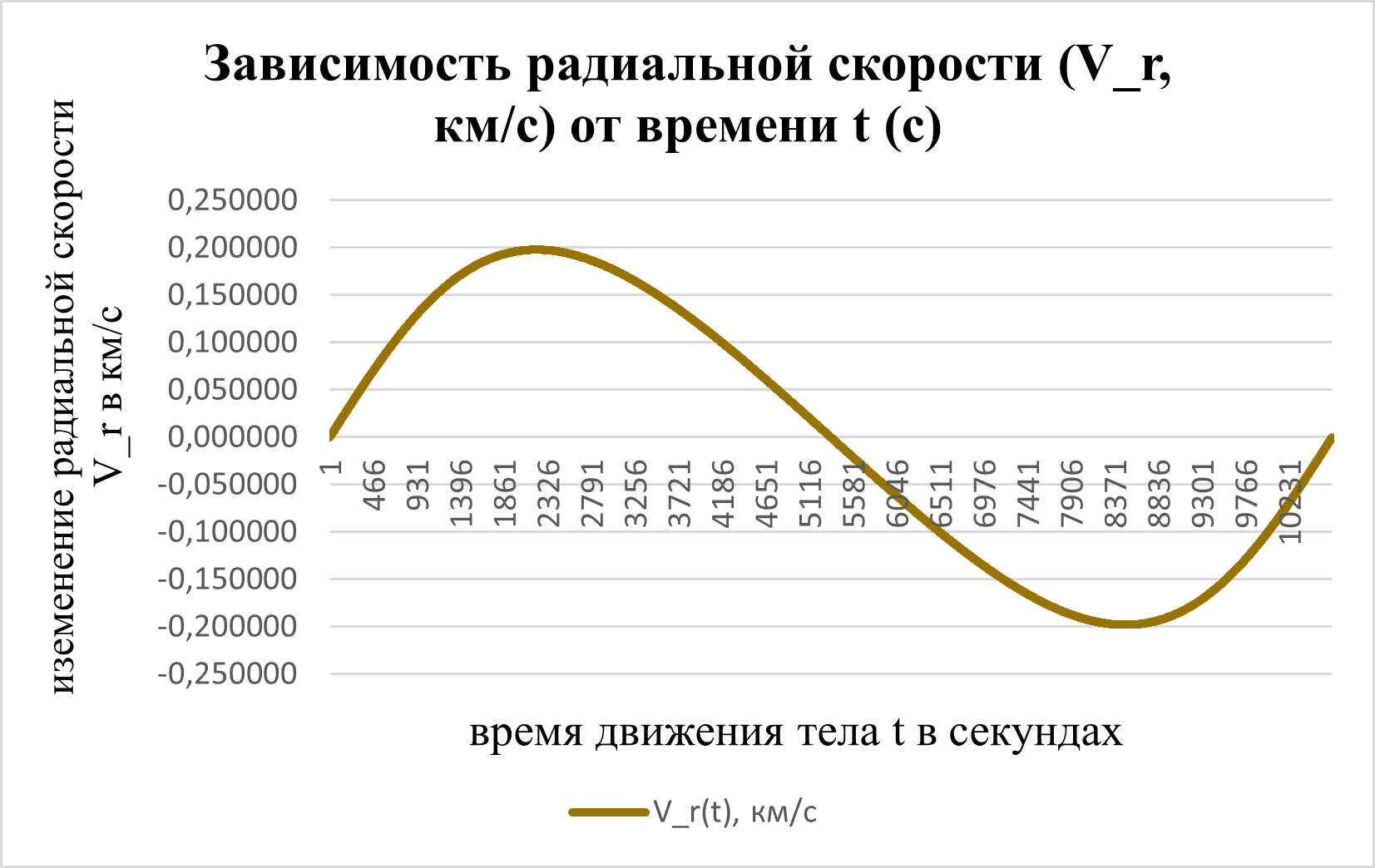


Рисунок 1. График зависимости Vr(t).

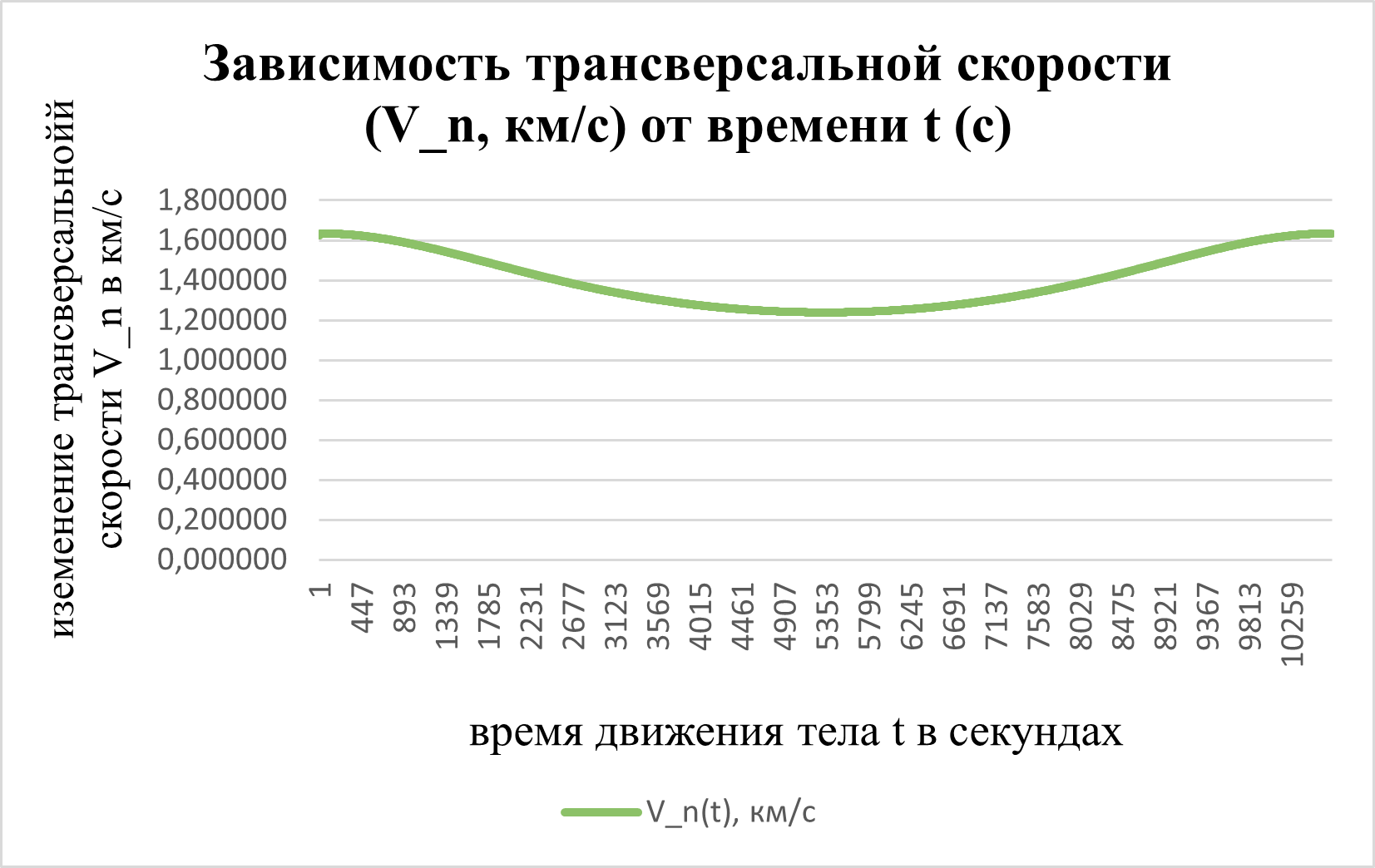


Рисунок 2. График зависимости Vn(t).

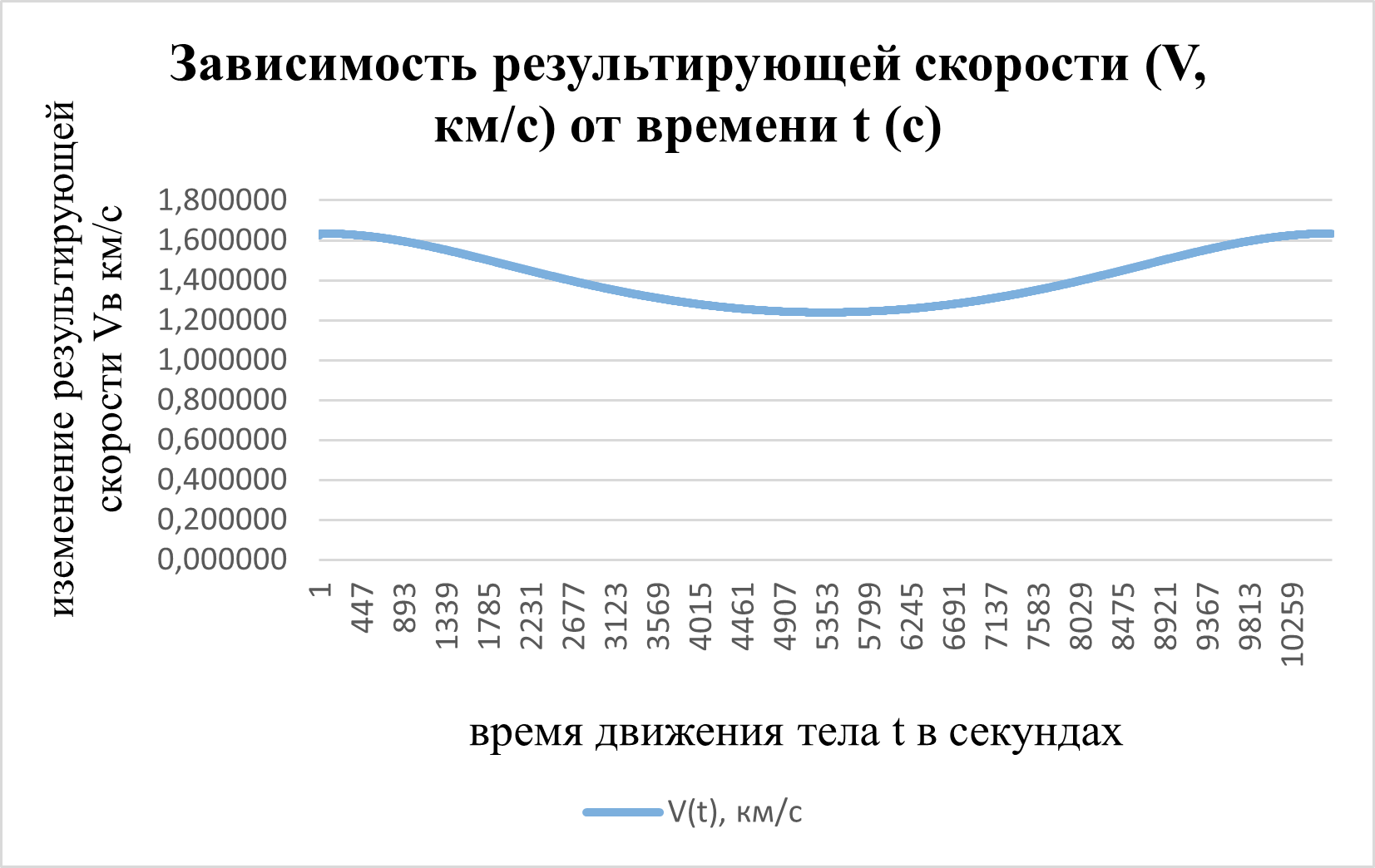


Рисунок 3. График зависимости V(t).

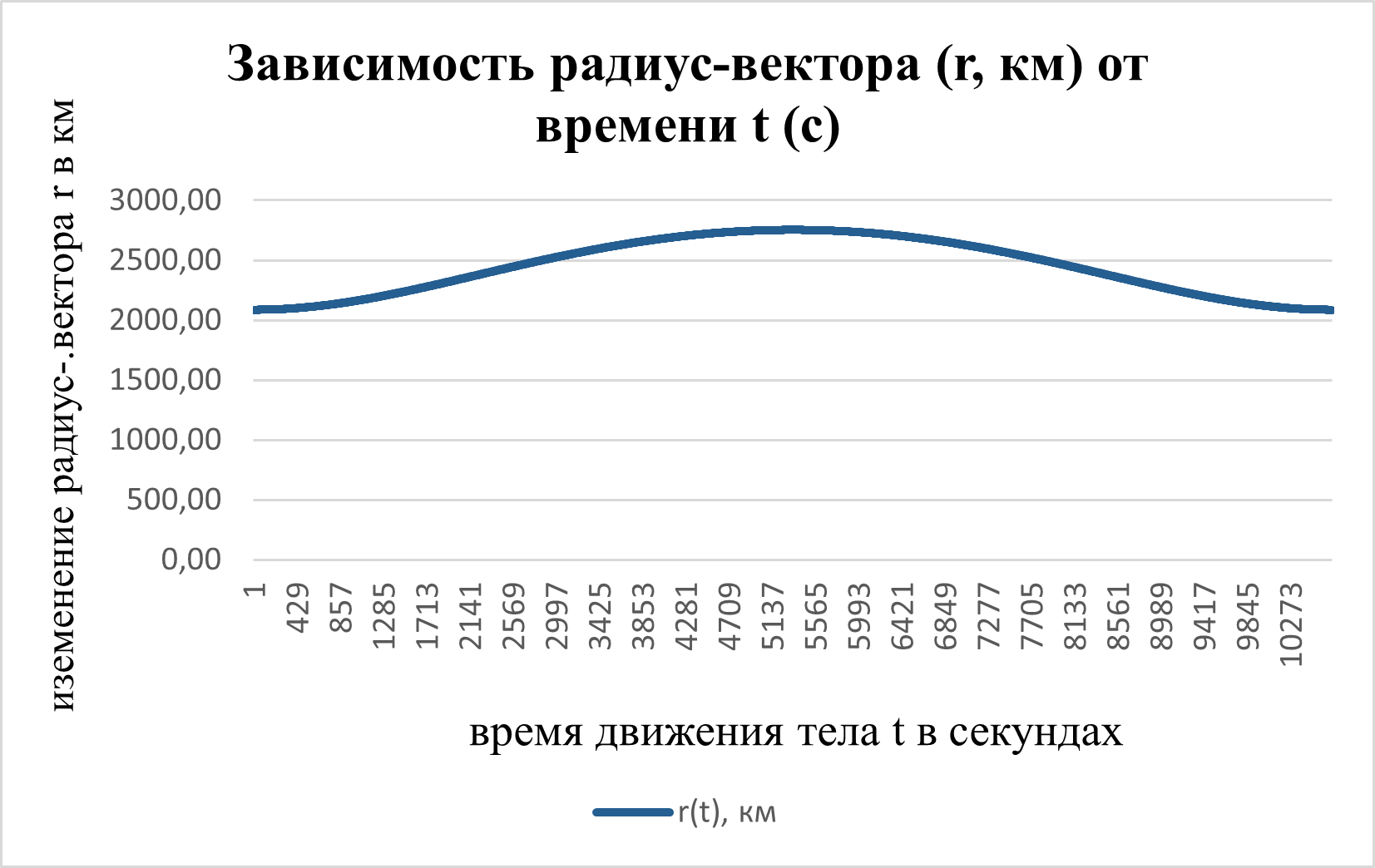


Рисунок 4. График зависимости r(t).

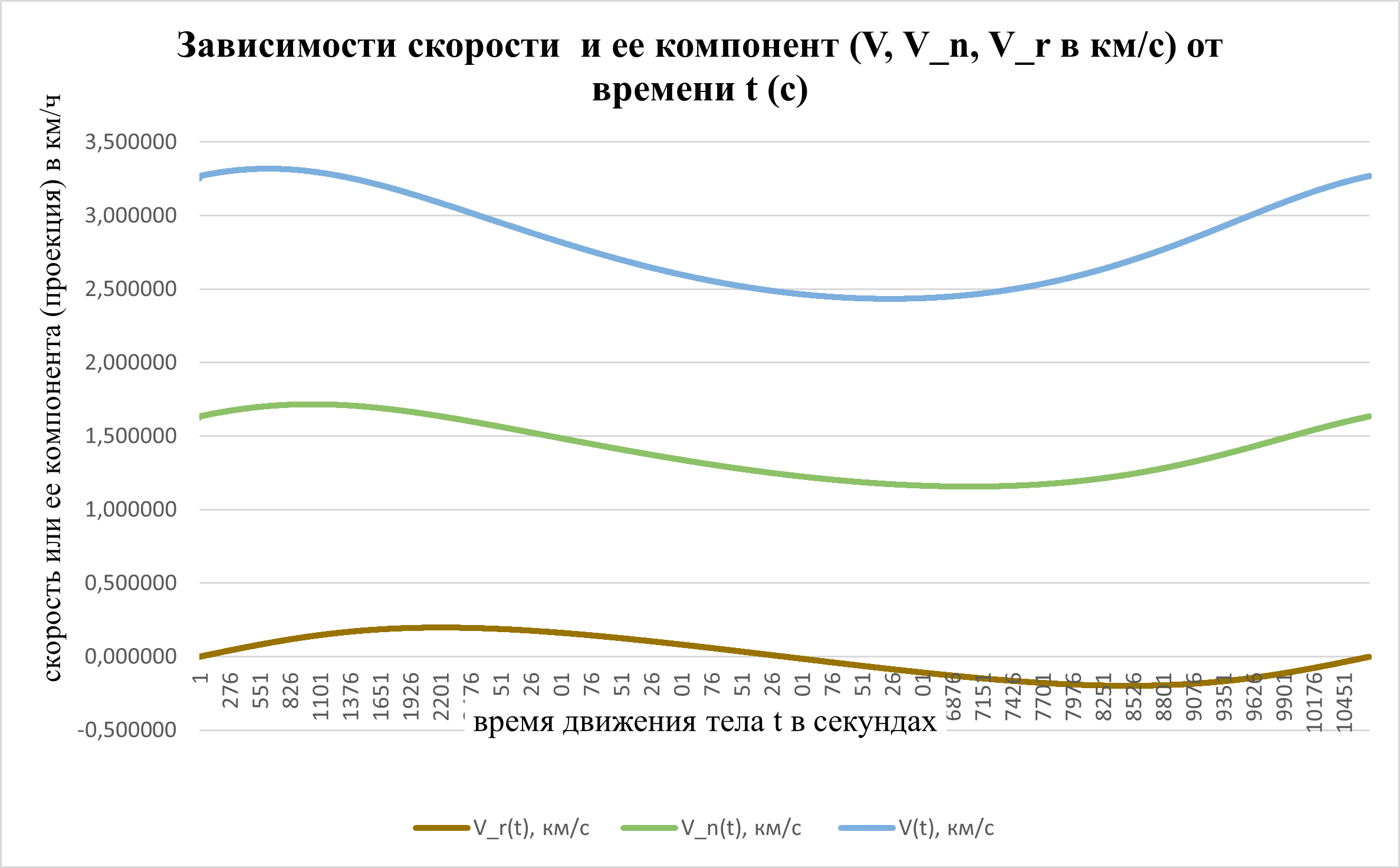


Рисунок 5. График зависимости Vr(t), Vn(t), V(t).

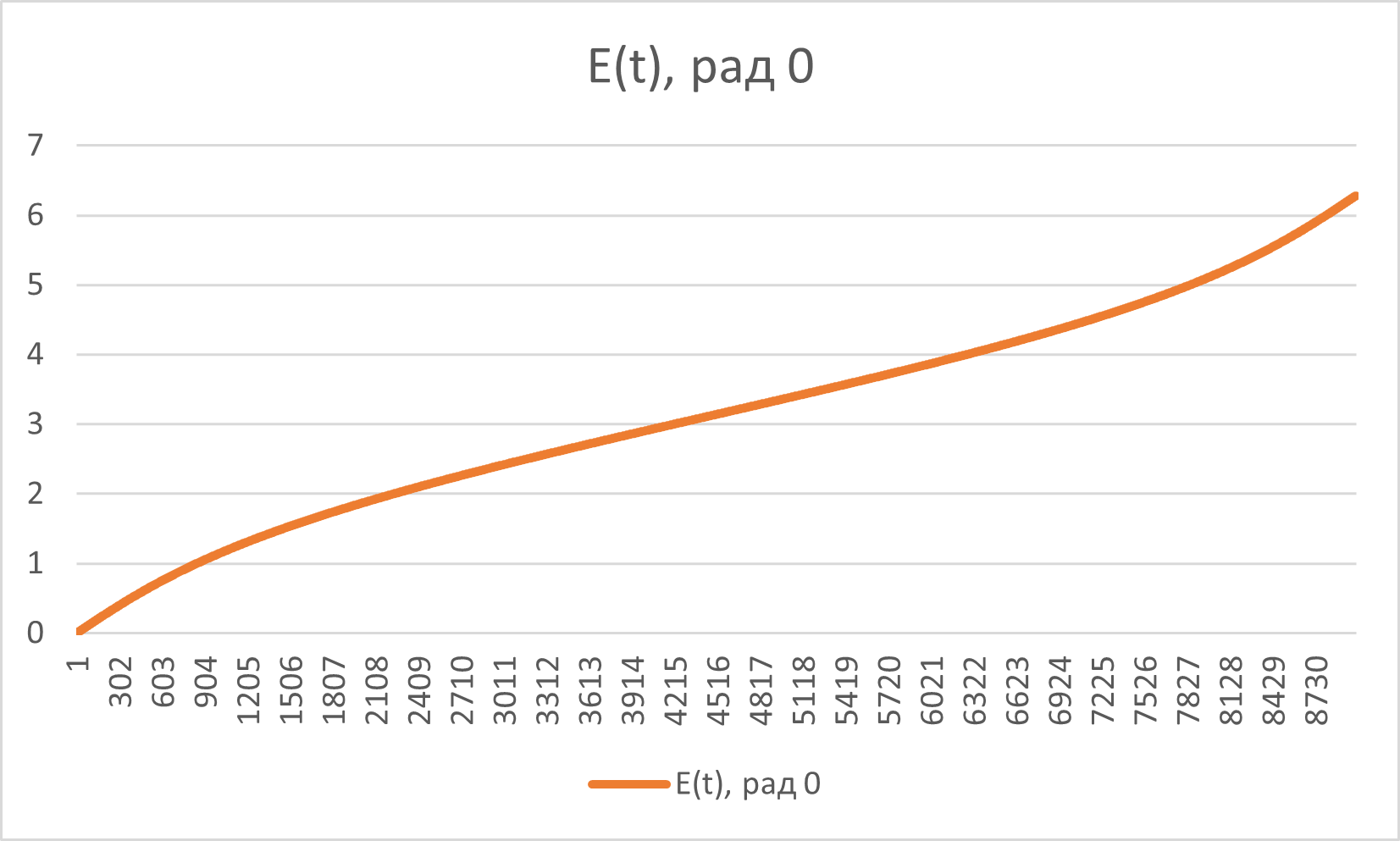


Рисунок 6. График зависимости E(t).

# Вывод

Трансверсальная составляющая скорости изменяется с 1.6349 км/с до 1,23894 км/c промежутке от 0 до 180 градусов км/с, c 1,23894 км/с до 1.6349 км/с на промежутке от 180 до 360 градусов.

Радиальная составляющая изменяется от 0 км/c до 0,19798 км/с на участке от 0 до (приблизительно) 81 градуса эксцентрической аномалии, затем значения падает до 0 в апоцентре. После прохождения апоцентра радиальная составляющая скорости идет со знаком минус, достигая минимального значения -0,19798 км/с в (приблизительно) 278 градусов E эксцентрической аномалии.

Максимального значения 1,6349 км/с V достигает в перицентре, минимального значения 1,23894 км/с – в апоцентре. Чем больше КА отдаляется от перицентра, тем меньше его скорость.

Своего минимального значения 2087 км радиус-вектор достигает в перицентре. Максимальное значение 2754 км r достигает в апоцентре.

Можно заметить, что чем больше радиус-вектор, тем меньше скорость КА.

# Приложение

1. Ссылка на GitHub: <https://github.com/A-no-maliya/CPLUSINRUDN/tree/master/laba_MKP_2/laba_MKP_3>
2. Ниже представлен исходный код программы на языке с++ с комментариями.

// подключаем библиотеки

#include <iostream>

#include <cmath>

#include <fstream>

#define PI 3.14159265358979323846

using namespace std;

// метод Ньютона для вычисления эксцентрической аномалии

double newton\_method(double M, double e, double epsilon = 1e-6)

{

double E = M; // Начальное приближение для E

for (int i = 0; i < 40; i++)

{

double f = E - e \* sin(E) - M; // Значение функции

double f\_prime = 1 + e \* cos(E); // Значение производной функции

E -= f / f\_prime; // Вычисление следующего приближения для E

if (abs(f) < epsilon) // Проверка условия сходимости

{

break;

}

}

return abs(E);

}

int main() {

// Параметры орбиты

double r\_a = 1017 + 1737; // км радиус апоцентра (h+r)

double r\_p = 350 + 1737; // км радиус перицентра (h+r)

double epsilon = 1e-6;

double GM\_m = 4902.80003; //в км

double a = (r\_a + r\_p) / 2; // большая полуось

double n = sqrt(GM\_m / pow(a, 3));// средняя угловая скорость

double T = 2 \* PI / n;

double e = ((r\_a - r\_p) / (2 \* a)); // эксцентриситет орбиты 0,48793

// Создание файла

ofstream fout1;

fout1.open("data\_for\_graphs.txt");

// Проверка успешного открытия файла

if (!fout1) {

cout << "Error opening file.";

return 0;

}

fout1 << "t, c\t";

fout1 << "M(t), рад\t";

fout1 << "E(t), рад\t";

fout1 << "Theta(t), рад\t" << endl;

fout1 << "r(t), км\t" << endl;

fout1 << "V\_r(t), м/с\t" << endl;

fout1 << "V\_n(t), м/с\t" << endl;

fout1 << "V(t), м/с\t" << endl; 1)

// Вычисление и запись значений аномалий в файл

Цикл вычисления M, E, , Vr, Vn, V для каждого момента t:

for (int t = 0; t <= T; t++) {

fout1 << t << "\t";

double M = n \* t; // Средняя аномалия

double E = newton\_method(M, e, epsilon); // Эксцентрическая аномалия

double true\_anomaly = atan(sqrt((1 + e) / (1 - e)) \* (tan(E / 2))) \* 2; // Истинная аномалия

if (true\_anomaly < 0)

true\_anomaly += 2 \* PI;

double p = a \* (1 - e \* e); // фокальное расстояние

double r = (p / (1 + (e \* cos(true\_anomaly)))); // вычисление радиус-вектора

double V\_n = (sqrt(GM\_m / p)) \* (1 + e \* cos(true\_anomaly)); // трансверсальная скорость

double V\_r = (sqrt(GM\_m / p)) \* (e \* sin(true\_anomaly)); //радиальная сокрость

double V = pow(((V\_r \* V\_r) + (V\_n \* V\_n)) , 0.5); // скорость тела

fout1 << M << "\t";

fout1 << E << "\t";

fout1 << true\_anomaly << "\t";

fout1 << r << "\t";

fout1 << V\_r << "\t";

fout1 << V\_n << "\t";

fout1 << V << endl;

}

fout1.close();

cout << "The data was successfully written to the 'data\_for\_graphs.txt' file.";

return 0;

}

1. Рисунок 7. Значение для 22 моментов времени t

