РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет/Институт: Инженерная академия\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра: Департамент механики и мехатроники\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КУРСОВАЯ РАБОТА

на тему:

**«Расчет параметров орбиты космического аппарата с помощью уравнения Кеплера»**

*01.03.02 «Прикладная математика и информатика»*

|  |
| --- |
| Выполнил:  Студент группы ИПБ-201  Студенческий билет №: 1032160153  Яблоновский Павел Алексеевич  Руководитель  д.т.н Ю.Н. Разумный |

Москва 2017

**Содержание**

Введение…………………………………………………………………………...3

1. Аналитический обзор…………………………………………………….….…4

1.1.Определение эллиптической орбиты……………………………….........4

1.2.Связь между временем и положением космического аппарата на орбите……………………………………………………………………….….6

1.3.Уравнение Кеплера для эллиптической орбиты……………….…..…..7

1.3.1.Решение уравнения Кеплера………………………...……….….….8

1. Код подпрограмм на языке C#………………………………………….....…9
2. Тестирование программы ......................………………………………...…..12

Список используемой литературы…………………………………………….18

**Введение**

Целью данной курсовой работы является расчет параметров орбиты космического аппарата. Чтобы выполнить данную работу воспользуемся уравнением Кеплера. Для решения поставленной задачи воспользуемся методом последовательных приближений (метод итерации), который является одним из наиболее распространенных и простейших численных методов решения уравнений.

Уравнение Кеплера не решается аналитически, то есть эксцентрическую аномалию невозможно явно представить, как функцию средней аномалии.

Так как координаты и их первые производные в эллиптическом движении могут быть представлены как явные функции эксцентрической аномалии, то найдя эксцентрическую аномалию как явную функцию времени, мы тем самым получим и явные выражения в зависимости от времени и всех других величин этого типа движения.

* 1. **Определение эллиптической орбиты**

Эллипс – это геометрическое место точек, для которых сумма расстояний от двух заданных точек (фокусов) есть величина постоянная. [**Эллипс**](http://www.mathelp.spb.ru/videomath8.htm)**-**это геометрическая фигура, которая ограничена кривой, заданной уравнением 

Для эллиптической орбиты имеем, что

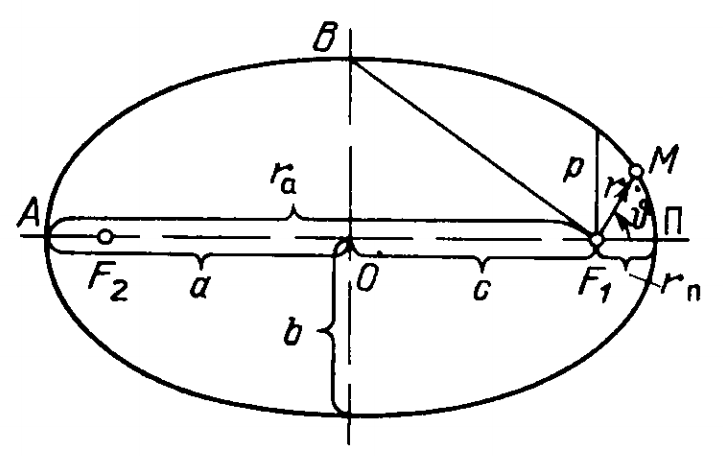
****

Рис. 1 Эллиптическая орбита

На рисунке 1 показаны основные параметры эллиптической орбиты: *a* – большая полуось, *b* – малая полуось,*p* – фокальный параметр, *c* – линейное фокусное расстояние.

Выразим большую полуось:

Подставим радиусы перицентра и апоцентра:

Выразим апогейное и перигейное расстояние через большую полуось:

По определению эллипса , где и –расстояние до одной и той же точки фокуса. Исходя из этого, получим относительный эксцентриситет *e*и малую полуось *b*:

* 1. **Связь между временем и положением космического аппарата на орбите**

Для определения закона орбитального движения космического аппарата необходимо установить зависимость параметров движения от времени.

Установим связь между временем и положением КА на орбите. Воспользуемся соотношением:

, где – время прохождения перигея

- уравнение Кеплера для Эллиптической орбиты.

**1.3 Уравнение Кеплера для эллиптической орбиты**

Уравнение Кеплера описывает движение тела по эллиптической орбите в задаче двух тел и имеет вид:

,

где *E* – эксцентрическая аномалия, *e*– эксцентриситет орбиты, *M* – средняя аномалия.

Уравнение Кеплера имеет одно решение:

Следовательно, является монотонной и имеет единственное решение.

* + 1. **Решение уравнения Кеплера**

Если время перелета спутника между двумя точками с известными величинами истинной аномалии вычисляется достаточно просто, то определение положения спутника в заданный момент времени, требует решения трансцендентного уравнения Кеплера или его аналогов для гиперболических и параболических.

Известно большое число алгоритмов, позволяющих найти приближенное решение уравнения Кеплера с любой степенью точности. При построении таких алгоритмов стараются учесть особенности задачи для упрощения вычислений. При малых значения эксцентриситета можно воспользоваться следующим алгоритмом:   
.

Для решения данной задачи воспользуемся **Методом последовательных приближений** – метод неподвижной точки, метод простой итерации – один из общих методов приближенного решения операторных уравнений. В ряде случаев хорошая сходимость построенных этим методом приближений позволяет применять его в практике вычислений.

Запишем уравнение Кеплера в виде . За нулевое приближение E(0) искомого корня E\* можно принять любое число, например 0 или M. Последовательность приближения E(n) к Е\* определяется следующим алгоритмом . Последовательность является сходящейся при любом выборе начального положения. Далее при рассмотрении ряда можем заметить, что

.

**2. Код подпрограммы на языке C#**

**Код, содержащийся в Form1.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO;

namespace Kepler

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

double mu = Convert.ToDouble(textBox1.Text);

double a = Convert.ToDouble(textBox2.Text);

double exs = Convert.ToDouble(textBox3.Text);

int tp = Convert.ToInt32(textBox4.Text);

int t = Convert.ToInt32(textBox5.Text);

double p = a \* (1 - exs \* exs);

double PI = 3.14159265;

double[] M = new double[t - tp+1];

double[] E = new double[t - tp+1];

double[] teta = new double[t - tp+1];

double[] radius = new double[t - tp+1];

System.IO.StreamWriter textFile = new System.IO.StreamWriter("text.txt");

textFile.WriteLine("Средняя аномалия");

for (int i = 0; i<=t-tp; i++)

{

M[i] = (Math.Sqrt(mu / (a \* a \* a)))\*(i+tp);

textFile.WriteLine("M["+i+"] = " + M[i]);

}

textFile.WriteLine(" ");

textFile.WriteLine("Эксцентрическая аномалия");

E[0] = M[0];

textFile.WriteLine("E[0] = " + E[0]);

for (int i = 1; i <= t - tp; i++)

{

E[i] = M[i] + exs \* Math.Sin(E[i - 1]\*PI/180);

textFile.WriteLine("E[" + i + "] = " + E[i]);

}

textFile.WriteLine(" ");

textFile.WriteLine("Истинная аномалия(Teta)");

for (int i = 0; i <= t-tp; i++)

{

teta[i] = 2 \* Math.Atan((Math.Sqrt((1 + exs) / (1 - exs)) \* Math.Tan(E[i]/2))\*PI/180);

textFile.WriteLine("teta[" + i + "] = " + teta[i]);

}

textFile.WriteLine(" ");

textFile.WriteLine("Радиус вектор");

for (int i = 0; i <= t - tp; i++)

{

radius[i] = a \* (1 - exs \* Math.Cos(E[i] \* PI / 180));

textFile.WriteLine("radius[" + i + "] = " + radius[i]);

}

textFile.Close();

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

}

private void textBox5\_TextChanged(object sender, EventArgs e)

{

}

private void label4\_Click(object sender, EventArgs e)

{

}

private void pictureBox1\_Click\_1(object sender, EventArgs e)

{

}

private void label3\_Click(object sender, EventArgs e)

{

}

private void pictureBox1\_Click\_2(object sender, EventArgs e)

{

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

}

private void label5\_Click(object sender, EventArgs e)

{

}

private void textBox6\_TextChanged(object sender, EventArgs e)

{

}

private void button2\_Click(object sender, EventArgs e)

{

// Displays an OpenFileDialog so the user can select a Cursor.

OpenFileDialog openFileDialog1 = new OpenFileDialog();

openFileDialog1.Filter = "Cursor Files|\*.txt";

openFileDialog1.Title = "Select a Cursor File";

// Show the Dialog.

// If the user clicked OK in the dialog and

// a .CUR file was selected, open it.

if (openFileDialog1.ShowDialog() == System.Windows.Forms.DialogResult.OK)

{

// Assign the cursor in the Stream to the Form's Cursor property.

this.Cursor = new Cursor(openFileDialog1.OpenFile());

}

}

}

}

**Код, содержащийся в Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Kepler

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

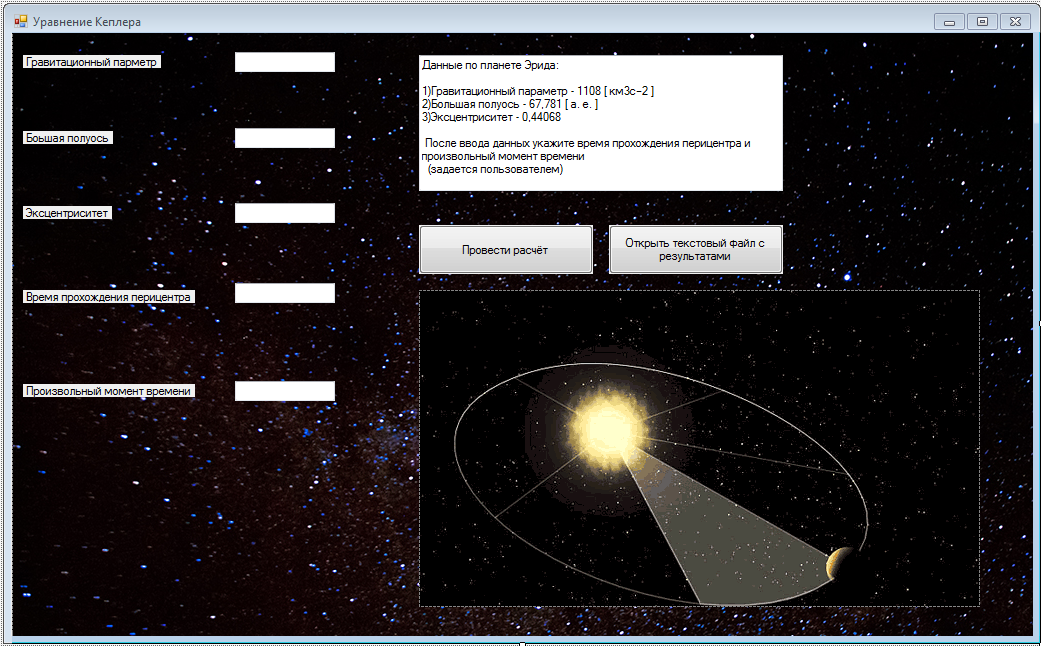
Application.Run(new Form1());

}

}

}

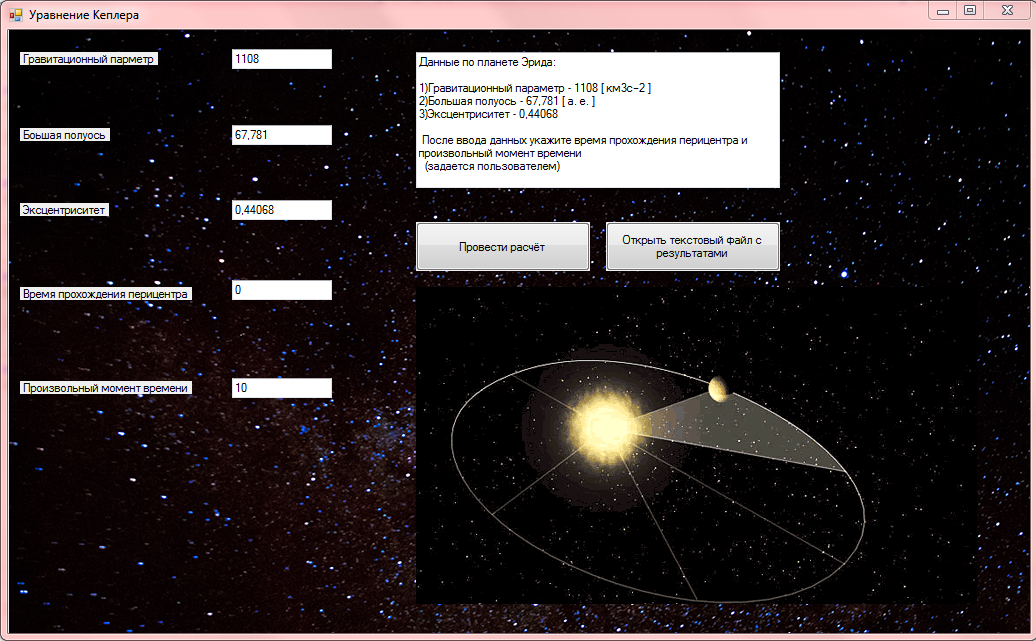
**Графически программа выглядит следующим образом:**

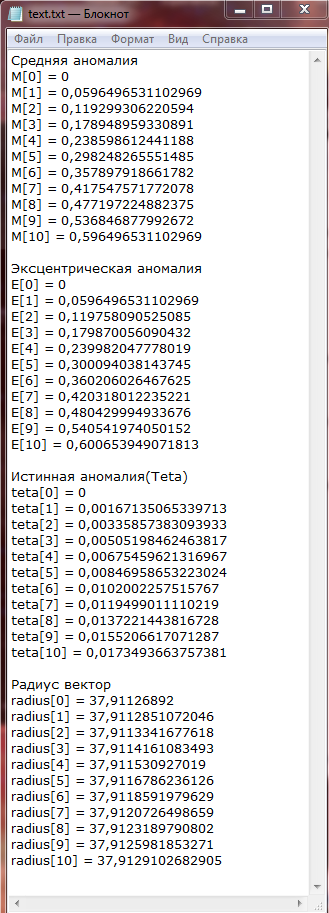


В данном интерфейсе представлены окна ввода данных, даны данный по планете, кнопка расчета, которая записывает результат в файл формата .txt и кнопка, которая позволяет открыть данный файл формата .txt. Так же представлено графическое изображение движения тела по эллиптической орбите в формате .gif.

**3.Тестирование программы на языке C#**

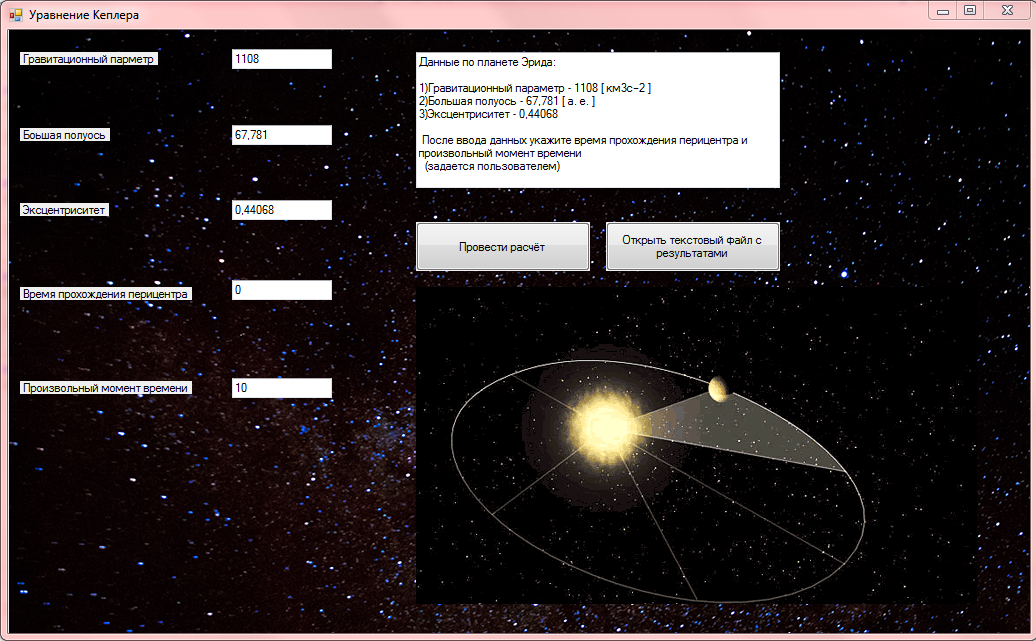
**Набор значений №1**

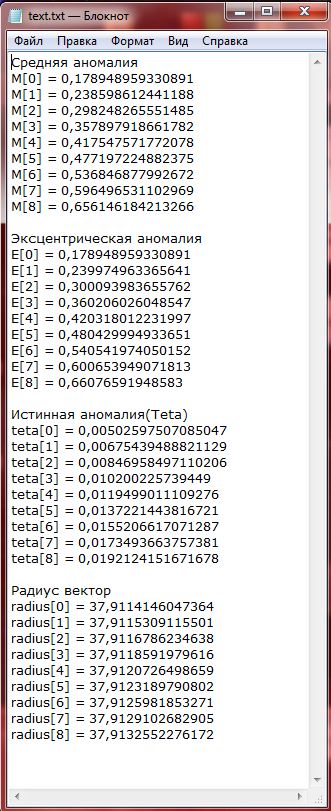
****

****

При значениях времени прохождения перицентра и произвольного момента времени, соответственно равными 0 и 10 программа записала в файл значения средней аномалии, эксцентрической аномалии, истинной аномалии и радиус вектора.

**Набор значений №2**





**Список используемой литературы.**

* Оливер Монтенбрук, Томас Пфлегер Астрономия на ПК.
* Г.Н.Дубошин Небесная механика. Основные задачи и методы.
* Соловьёв В.А., Лысенко Л.Н., Любинский В.Е. Управление космическими полётами. В 2 ч.Изд-во МГТУ им. Н.Э. Баумана
* Охоцимский Д.Е., Сихарулидзе Ю.Г. Основы механики космического полёта.
* Российская академия наук, институт прикладной математики им. М.В.Келдыша. Механика космического полета. Орбитальное движение. *Учебно-методическое* пособие. Профессор С.А. Мирер
* Лекции с курса теории космического полета