

Всероссийский форум научной молодежи
«Шаг в будущее»

Научно-исследовательская работа
Обобщённый анализ кометных орбит

Выполнил: Гига Максим Янович, 8 класс
МАУ ДО Дом детского творчества
города Звенигород

Научный руководитель:
Вибе Анжелика Анатольевна,
педагог дополнительного образования, МАУ ДО
Дом детского творчества города Звенигород

Звенигород, 2020

Обобщённый анализ кометных орбит

Гига Максим Янович

Россия, Московская область, г. Звенигород

Муниципальное автономное учреждение дополнительного образования
Дом детского творчества города Звенигород, 8 класс

Введение

Кометы — это самые необычные тела Солнечной системы, привлекающие внимание человека с глубокой древности. Обращающиеся вокруг Солнца по сильно вытянутым орбитам и появляющиеся, как правило, неожиданно, кометы имеют главную особенность — при сближении с Солнцем у них появляется хвост, отличающий их от остальных малых тел.

Сегодня астрономам известно свыше 4000 комет [1]. Впервые период у кометы вычислил Эдмунд Галлей в 1716 г., это была яркая комета, получившая его имя. В зависимости от периода обращения вокруг Солнца кометы делятся на короткопериодические и долгопериодические. К короткопериодическим относятся кометы, период обращения которых вокруг Солнца составляет меньше 200 лет, и те, у которых наблюдалось 2 и более прохождения перигелия. Короткопериодические кометы приходят из области планет-гигантов, двигаясь в прямом направлении (как и планеты) по орбитам, слегка наклонённым к эклиптике [2].

К долгопериодическим относятся кометы, чей период обращения составляет больше 200 лет, или если они наблюдались возле Солнца только один раз. Долгопериодические кометы прилетают из облака Оорта — гипотетической области, расстояние до внешних границ которой от Солнца составляет 50-100 тысяч астрономических единиц [3]. Орбиты долгопериодических комет наклонены к эклиптике под разными углами, а обращение вокруг Солнца может быть как в прямом, так и в обратном направлении. К долгопериодическим кометам относятся и кометы, орбиты которых незамкнуты.

Цель данного исследования — построение пространственного представления о положении кометных орбит в Солнечной системе.

Задачи исследования — используя базу данных элементов орбит комет, построить при помощи компьютерной программы распределение орбит комет в Солнечной системе по формам, наклонениям, семействам, выяснить зависи-

мость между наклоном и эксцентриситетом орбит, влияние Юпитера на кометы, предложить объяснения выявленным закономерностям.

Ход работы

- 1) Изучение информации о движении комет, факторах, влияющих на их движение, об исследованиях орбит комет и их классификации.
- 2) Выбор способа обработки базы данных элементов орбит комет, написание программы «Кометные орбиты» для графического построения орбит.
- 3) Построение положения кометных орбит в Солнечной системе: распределение по периодам, семействам, наклонам, определение зависимости между наклоном и эксцентриситетом орбит, выяснение влияния Юпитера.
- 4) Анализ результатов, поиск закономерностей полученного распределения, объяснение причин, вызывающих подобные закономерности.

База данных орбит комет

При выполнении исследования использовалась база данных «JPL Solar System Dynamics» Small-Body Orbital Elements (Comets)» (<https://ssd.jpl.nasa.gov/dat/ELEMENTS.COMET>), содержащая элементы орбит 3586 комет. Следует отметить, что информация, представленная в любой базе данных, через некоторое время может оказаться неактуальной. Это связано с тем, что орбиты комет изменяются в результате притяжения крупных планет, испарения (и/или реактивного движения), а также могут распасться на части. Так, например, в данной базе данных содержатся элементы орбит фрагментов комета Шумейкеров-Леви-9, упавшей на Юпитер в 1994 г.

Орбиты комет

Согласно законам механики, движение космического тела (кометы) под действием гравитационного притяжения другого тела (Солнца) происходит по одному из конических сечений — эллипсу (частный случай окружность) или гиперболе (частный случай парабола) [2]. Полное описание движения кометы в гравитационном поле Солнца достигается заданием шести величин — элементов орбиты, характеризующих её начальное положение и скорость движения.

В зависимости от афелийного расстояния (самое большое расстояние тела от Солнца) кометы можно отнести к семействам планет. Большинство самых короткопериодических комет (период 3-10 лет) образуют *семейство Юпитера*.

ра. Немного малочисленнее семейства Сатурна с периодами 10-20 лет, семейства Урана (период 28-40 лет) и семейства Нептуна (период 58-120 лет). Комета Галлея сначала относилась к семейству Нептуна, теперь считается, что существует семейство кометы Галлея. Кометы, орбитальный период обращения которых, как у кометы Галлея, составляет от 20 до 200 лет, называются кометами *галлеевского типа* [4]. Кометы, приближающиеся к Солнцу в перигелии на очень близкое расстояние (сотые и тысячные доли астрономической единицы) называются *околосолнечными*. Среди них выделяется несколько семейств — Крейца, Мейера, Марседена, Крахта и др. Наиболее многочисленным из них является семейство *околосолнечных комет Крейца*, названное в честь немецкого астронома Г. Крейца, который впервые показал их взаимосвязь. Считается, что все они являются частями одной большой кометы, которая разрушилась несколько столетий назад [5].

Элементы орбиты

Элементы орбиты — это набор параметров, задающих размеры и форму орбиты (траектории движения) небесного тела, расположение орбиты в пространстве и положения небесного тела на орбите.

В используемой базе данных приведены следующие элементы орбит:

- *эпоха* — модифицированная юлианская дата;
- *перигелийное расстояние q* — минимальное расстояние от Солнца;
- *эксцентриситет e* — степень вытянутости орбиты;
- *наклонение i* — угол между плоскостью орбиты кометы и плоскостью эклиптики;
- *аргумент перицентра w* — угол между направлениями из притягивающего центра (Солнца) на восходящий узел орбиты и на перицентр;
- *долгота восходящего узла $Node$* — угол между направлением на точку отсчёта системы координат и направлением на точку восходящего узла орбиты;
- *время T_p* — момент прохождения перигелия.

В зависимости от эксцентриситета орбиты бывают: *эллиптические* при $0 < e < 1$ (в частном случае, *круговые* при $e = 0$) и *гиперболические* при $e > 1$ (в частном случае, *параболические* при $e = 1$) [2].

Сортировка орбит комет

Для сортировки и обработки базы данных «JPL Solar System Dynamics» Small-Body Orbital Elements (Comets)» была написана компьютерная программа «Кометные орбиты» (см. Приложение 1) на языке программирования

java. Задачи программы — выбор из базы данных элементов орбит, вычисление недостающих элементов, сортировка комет по периодам, наклонениям, форме, подсчёт количества объектов для заданного семейства, вывод результата.

1) Сортировка комет по периоду. Всего было обработано 3586 объектов, из которых 2191 имеют гиперболические (и параболические) орбиты — это долгопериодические кометы, а 1239 обладают эллиптическими орбитами — короткопериодические кометы. Результат сортировки показан в Таблице 1.

Таблица 1. Распределение комет по периодам

| Орбита | Период | Количество |
|----------------------------------|----------------------|------------|
| Эллиптическая | короткопериодическая | 1239 |
| Гиперболическая (параболическая) | долгопериодическая | 2347 |
| Итого | | 3586 |

2) Сортировка комет по семействам. В данном исследовании используются только короткопериодические кометы. Результат сортировки показан в Таблице 2.

Таблица 2. Распределение комет по семействам

| Семейство комет | Афелийное расстояние | Количество |
|-------------------|----------------------|------------|
| Меркурия | 0,25 – 0,55 | 0 |
| Венеры | 0,56 – 0,85 | 0 |
| Земли | 0,86 – 1,25 | 0 |
| Марса | 1,26 – 1,75 | 0 |
| Юпитера | 1,76 – 7,30 | 487 |
| Сатурна | 7,31 – 14,60 | 214 |
| Урана | 14,61 – 25,50 | 42 |
| Нептуна | 25,51 – 35,00 | 36 |
| галлеевского типа | 35,01 – 100000 | 460 |
| Итого | | 1239 |

Как видно из таблицы, планеты земной группы совсем не имеют своих комет. Больше всего комет в семействе Юпитера — 487, за пределами орбиты Нептуна насчитывается 460 комет семейства галлеевского типа.

3) Сортировка по направлению движения. В зависимости от наклонения орбиты, движение кометы является прямым, если $0^\circ < i < 90^\circ$ и обратным, если $90^\circ < i < 180^\circ$. Результат сортировки показан в Таблице 3.

Таблица 3. Количество комет с прямым и обратным движением для разных семейств

| Семейство комет | Общее количество | Прямое движение (кол-во/%) | Обратное движение (кол-во/%) |
|--------------------|------------------|----------------------------|------------------------------|
| Юпитера | 487 | 487/100% | 0 |
| Сатурна | 214 | 207/96,7% | 7/3,3% |
| Урана | 42 | 33/78,6% | 9/21,4% |
| Нептуна | 36 | 28/77,8% | 8/22,2% |
| галлеевского типа | 460 | 256/55,6% | 204/44,4%. |
| долгопериодические | 2347 | 562/23,9% | 1785/76,1% |
| Итого | 3586 | 1573/43,9% | 2013/56,1% |

4) Сортировка семейства околосолнечных комет. В используемой базе данных околосолнечных комет насчитывается 1495. Эти кометы движутся, в основном, по параболическим орбитам. Наклонения орбит этих комет показаны в *Приложении 2* и на рисунке 1 а), орбиты комет в *Приложении 3* и на рисунке 1 б).

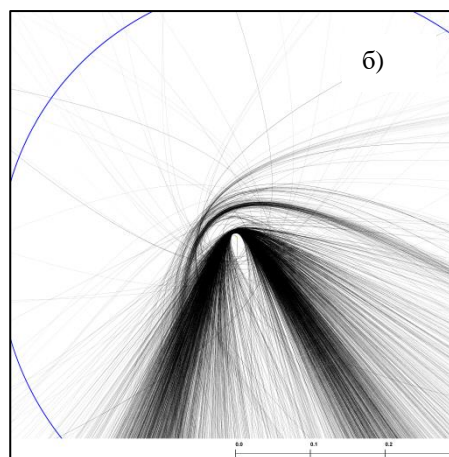
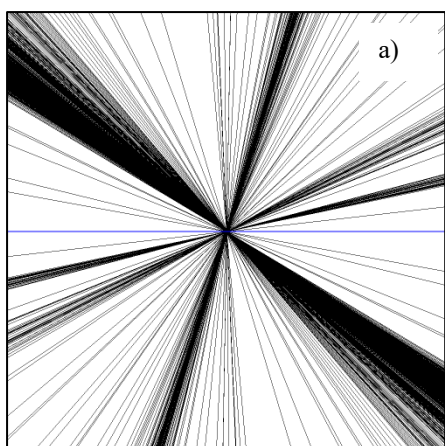


Рис. 1. Орбиты семейства околосолнечных комет: а) наклонения (голубая линия — плоскость эклиптики); б) положения орбит (голубая линия — орбита Меркурия)

5) Зависимость наклонения орбиты от эксцентриситета. В этом исследовании проверялось наличие или отсутствие зависимости эксцентриситета орбиты от её наклонения. Результат показан в *Приложении 4*.

б) Положение комет по отношению к Юпитеру. Планета Юпитер очень массивна и может притягивать к себе кометы, не только захватывая их в своё семейство, но и выстраивая их около себя. Если долготы восходящего узла орбит комет семейства Юпитера лежат в пределах от $90,5^\circ$ до $110,5^\circ$ по отношению к долготе восходящего узла самого Юпитера ($100,5^\circ$), то кометы захвачены Юпитером. Исследование показывает, что количество комет с такими долготами составляет 23 (4,7%) из 487 комет семейства.

Результаты и обсуждение

Количество комет в семействах различно, причина этого — разная масса планет, например, мощное гравитационное влияние на кометы с относительно малыми массами производит Юпитер, поэтому его семейство самое многочисленное. Суммарная масса Урана и Нептуна всего 32 массы Земли [6], поэтому и семейства у них малочисленны.

Для лучшего понимания того, какое наклонение характерно для орбит комет, было сделано графическое представление их наклонений для разных семейств (см. *Приложение 2*), например, для семейств Юпитера и Сатурна наклонения орбит показаны на рисунке 2.

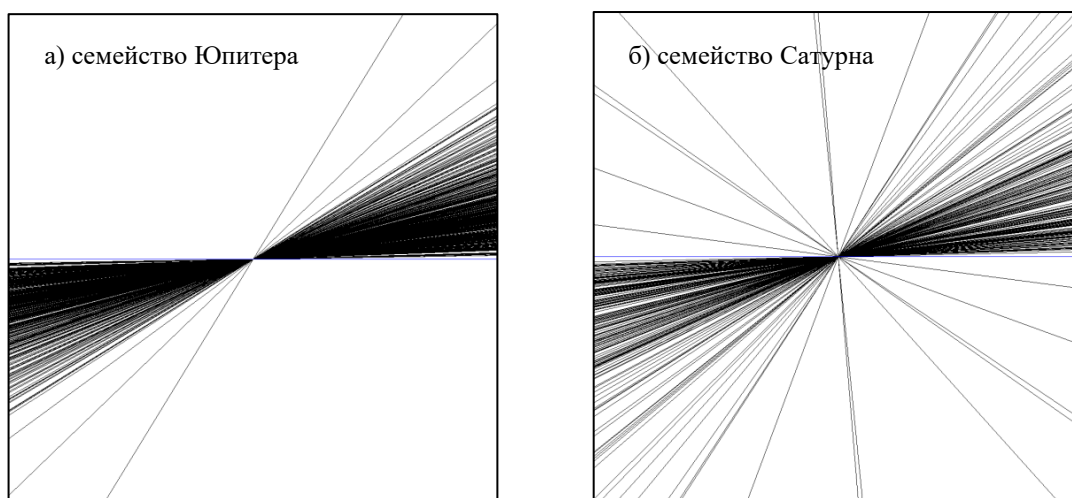


Рис. 2. Наклонения орбит комет семейств а) Юпитера и б) Сатурна

Полученные графические представления говорят о том, что при удалении от Солнца появляется всё больше комет с обратным движением. Так, в семей-

стве Юпитера нет комет с обратным движением, большая их часть движется по орбитам с наклоном от 0° до 40° . Если рассматривать объекты за орбитой Нептуна, то среди них уже 204 кометы из 460 обладают обратным движением, что составляет 44,3%. Среди околосолнечных комет выделяется три предпочтительных наклонения орбит и три предпочтительных направлений движения комет, что говорит о существовании, как минимум, трёх семейств околосолнечных комет (например, семейства Крейца и др.).

Для полного пространственного представления движения комет в Солнечной системе было сделано графическое представление форм орбит для разных семейств (см. Приложение 3). Например, на рисунке 3 показаны орбиты комет семейств Юпитера, Сатурна, Нептуна и галлеевского типа.

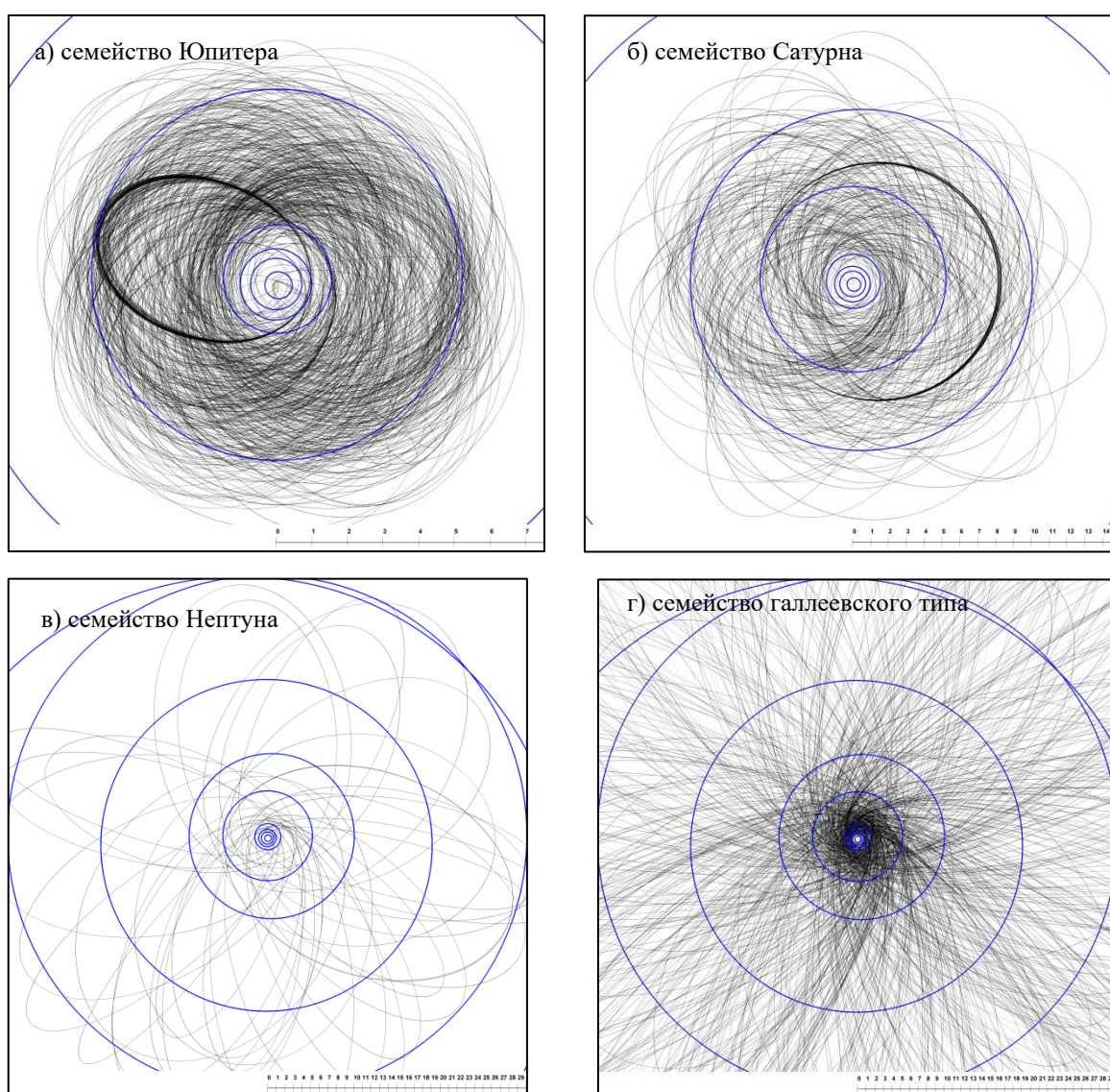


Рис. 3. Орбиты комет семейств Юпитера (тёмные места — большая плотность орбит комет Швассмана-Вахмана), Сатурна (орбиты фрагментов кометы Шумейкерв-Леви-9), Нептуна, галлеевского типа. Синие окружности — орбиты планет

Как видно из рисунков, кометы семейства Юпитера движутся по коротким эллиптическим орбитам, близким к окружностям, а орбиты комет галлеевского типа имеют очень большую вытянутость. Большая концентрация кометных орбит в правой части на рисунке 3 б) может означать либо орбиты фрагментов большой кометы, которые со временем сильно разошлись, либо под воздействием притяжения сразу двух самых крупных планет Юпитера и Сатурна (их суммарная масса составляет 413 масс Земли [6]), образовалось *«юпитерианско-сатурнианское подсемейство»* комет семейства Юпитера. При этом сам Юпитер не создаёт вокруг себя кометного «окружения», а фелии орбит комет его семейства равномерно распределились в области его орбиты.

Кроме того, зависимость наклона орбиты от её эксцентриситета, вопреки ожиданиям (предполагалось, что чем больше эксцентриситет у орбит, тем больше должно быть наклонение), не обнаружена, оказалось, что вытянутые и не очень вытянутые орбиты имеют различные наклонения. Это может означать, что кометы, пришедшие в центральную часть Солнечной системы из облака Оорта с разными наклонениями, со временем могут уменьшать вытянутость своей орбиты до почти круговой, например, в результате испарения, а вот изменить наклонение своей орбиты им гораздо труднее.

Результаты исследования говорят о том, что такие особенности движения комет, возможно, связаны с их разным происхождением. Кометы семейства Юпитера либо образовались неподалёку от него, либо давно уже были им захвачены, поэтому имеют стабильные эллиптические орбиты с небольшим наклонением и эксцентриситетом. А кометы семейства галлеевского типа попали туда (были гравитационно захвачены) из далёкого сферического облака Оорта, т.е. из всевозможных направлений, поэтому имеют разные наклонения и большой эксцентриситет.

В целом, анализ орбит комет показывает, что, несмотря на постоянно изменяющуюся траекторию движения под воздействием гравитационного притяжения планет, реактивного движения самих комет, испарения газов и пыли (и, следовательно, изменения массы кометы и уменьшения её орбиты), в центральной части Солнечной системы они «ведут себя» более-менее единообразно и упорядочено, а на далёких окраинах — совершенно непредсказуемо и хаотично.

Заключение

С помощью разработанной компьютерной программы «Кометные орбиты» проанализирована база данных «JPL Solar System Dynamics» Small-Body Orbital Elements (Comets)», построено распределение около 3586 комет по периодам, семействам, формам, по направлению движения, не выявлена связь между наклоном и эксцентриситетом орбит и не обнаружено влияние Юпитера на своё семейство, построено графическое представление наклонов и положений орбит комет разных семейств в Солнечной системе, предложены объяснения причин такого распределения.

Исследование и глубокое изучение комет актуально всегда, поскольку их физические свойства, движение и происхождение помогут лучше понять эволюцию Солнечной системы в целом. Кометы могли возникнуть и как побочный продукт при формировании планет, и вследствие других процессов в протосолнечном газопылевом облаке. Бесспорно, что они зародились в Солнечной системе, мало эволюционировали и поэтому несут в себе информацию о её начальных условиях. Знания об их движении и составе могут внести весомый вклад в решение фундаментальных вопросов о происхождении Солнечной системы, планетных систем в Галактике, а также о зарождении жизни на Земле и в целом во Вселенной.

Список источников и литературы

1. Солнечная система / ред.-сост. В.Г. Сурдин. — М. ФИЗМАТЛИТ, 2008. — 400 с. — (Астрономия и астрофизика).
2. Марочник Л.С., Свидание с кометой. — Наука. Гл. ред. физ.-мат. лит., 1985. — 208 с. (Библиотечка «Квант». Вып. 47.)
3. Маров, М.Я. Космос: От Солнечной системы вглубь Вселенной / М.Я. Маров. — М.: ФИЗМАТЛИТ, 2016. — 536 с.
4. Бирюков Е.Е. Захват комет из облака Оорта на орбиты галлеевского типа и орбиты семейства Юпитера. — Астрономический вестник, 2007, Т. 41, № 3, С. 232-240.
5. Малые тела Солнечной системы. Околосолнечные кометы. [Электронный ресурс] — Режим доступа: <http://galspace.spb.ru/index385.html>
6. Астрономия: век XXI / Ред.-сост. В.Г. Сурдин. — Фрязино: «Век 2», 2008. — 2-е изд., испр. и доп. — 608 с.: ил.

Программа «Кометные орбиты»

```

package com.company;

import javax.imageio.ImageIO;
import java.awt.*;
import java.awt.geom.AffineTransform;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.nio.charset.StandardCharsets;
import java.util.ArrayList;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) throws FileNotFoundException, IOException {
        final String fileName = "D:\\file3.txt";
        String[] list = readUsingScanner(fileName).trim().split("\\s+");
        ArrayList<Double>[][] data1 = new ArrayList[11][7];
        double[] distances = {0, 0.45, 0.85, 1.25, 1.7, 7.3, 14.6, 25.5, 35, 100000};
        double[][] results = new double[11][3];
        Graphic g = new Graphic(1000, 1000);
        Graphic2 g2 = new Graphic2();
        Graphic3 g3 = new Graphic3();
        for (int i = 0; i < 11; i++) {
            for (int i2 = 0; i2 < 7; i2++) {
                data1[i][i2] = new ArrayList<>();
            }
        }
        System.out.println(list.length / 5);
        for (int i = 0; i < list.length; i += 5) {
            if (Double.parseDouble(list[i + 1]) < 0.999) {
                double periDist = Double.parseDouble(list[i]);
                double excentricity = Double.parseDouble(list[i + 1]);
                double inclination = Double.parseDouble(list[i + 2]);
                double longitude = Double.parseDouble(list[i + 4]);
                double periapsis = Double.parseDouble(list[i + 3]);
                double majorAxis = periDist / (1 - excentricity);
                double apheDist = majorAxis * (1 + excentricity);
                for (int i2 = 0; i2 < distances.length - 1; i2++) {
                    if (apheDist > distances[i2] && apheDist <= distances[i2 + 1]) {
                        data1[i2][0].add(periDist);
                        data1[i2][1].add(excentricity);
                        data1[i2][2].add(inclination);
                        data1[i2][3].add(longitude);
                        data1[i2][4].add(majorAxis);
                        data1[i2][5].add(apheDist);
                        data1[i2][6].add(periapsis);
                        results[i2][0] = data1[i2][0].size();
                    }
                }
            } else {
                double inclination = Double.parseDouble(list[i + 2]);
                data1[10][2].add(inclination);
            }
        }
        System.out.println("hello");
        for (int i = 0; i < 11; i++) {
            int countInclination = 0;
            for (int i2 = 0; i2 < data1[i][2].size(); i2++) {
                if (data1[i][2].get(i2) < 90) {
                    countInclination++;
                }
            }
            results[i][1] = countInclination;
        }
    }
}

```

```

    }
    System.out.println(data1[4][1].size());
    System.out.println(results[4][1]);
    for (int i = 0; i < distances.length - 1; i++) {
        System.out.println("from: " + distances[i] + " to: " + distances[i + 1]);
        System.out.println("count of comets: " + results[i][0]);
        System.out.println("count of comets with right inclination: " + re-
sults[i][1]);
        System.out.println("-----");
        System.out.println();
        double[] array2 = new double[data1[i][2].size()];
        double[] arrayPeriapsis = new double[data1[i][2].size()];
        double[] arrayEccentricity = new double[data1[i][2].size()];
        double[] arrayPeridist = new double[data1[i][2].size()];
        for (int i2 = 0; i2 < data1[i][2].size(); i2++) {
            array2[i2] = data1[i][2].get(i2);
            arrayPeriapsis[i2] = data1[i][6].get(i2);
            arrayEccentricity[i2] = data1[i][1].get(i2);
            arrayPeridist[i2] = data1[i][0].get(i2);
        }
        if(i==4){
g3.drawGraphic(arrayEccentricity,array2,"D:\\programing\\java\\newEllipses_ItoE"+i+".p
ng");
        }
        //g.draw(array2, "D:\\programing\\java\\new"+i+".png",);
        /*if(i==1){
            g2.draw(arrayPeriapsis, arrayEccentricity, arrayPeridist,
"D:\\programing\\java\\newEllipses"+i+".png",400.0);
        }*/
        /*if(i==4){
            g2.draw(arrayPeriapsis, arrayEccentricity, arrayPeridist,
"D:\\programing\\java\\newEllipsesRed"+i+".png",400.0);
        }else if(i==5){
            g2.draw(arrayPeriapsis, arrayEccentricity, arrayPeridist,
"D:\\programing\\java\\newEllipsesRed"+i+".png",200.0);
        }else{
            g2.draw(arrayPeriapsis, arrayEccentricity, arrayPeridist,
"D:\\programing\\java\\newEllipsesRed"+i+".png",100.0);
        }*/
    }
    System.out.println(data1[10][2].size());

//System.out.println(data1[0][2].size()+data1[1][2].size()+data1[2][2].size()+data1[3]
[2].size()+data1[4][2].size()+data1[5][2].size()+data1[6][2].size()+data1[7][2].size()
+data1[8][2].size()+data1[9][2].size()+data1[10][2].size());
    System.out.println(results[10][1]);
    /*double[] periapsis = {200};
    double[] eccentricity = {0.5};
    double[] peridist = {100};
    g2.draw(periapsis, eccentricity, peridist, "D:\\picture.png", 1);*/
}

    private static String readUsingScanner(String fileName) throws IOException {
        String data;
        try (Scanner scanner = new Scanner(new File(fileName), Standard-
Charsets.UTF_8.name())) {
            data = scanner.useDelimiter("\\A").next();
        }
        return data;
    }
}

class Graphic {
    final int linesLenght = 5000;
    final double startAngle = 90;
    int width = 1000;
    private int height;
    private final int[] centerPoint = {width / 2, height / 2};
}

```

```

double angleSet[];

Graphic(int width, int height) {
    this.width = width;
    this.height = height;
}

public void draw(double[] angle42, String name) {
    this.angleSet = angle42;
    BufferedImage bi = new BufferedImage(1000, 1000, BufferedImage.TYPE_INT_ARGB);
    Graphics2D g = bi.createGraphics();
    g.setColor(Color.WHITE);
    g.fillRect(0, 0, 10000, 10000);
    g.setColor(Color.black);
    for (int i = 0; i < angleSet.length; i++) {
        double angle = (startAngle + angleSet[i]) * Math.PI / 180;
        double angle2 = (startAngle + angleSet[i] + 180) * Math.PI / 180;
        int endX1 = (int) Math.round(centerPoint[0] + linesLenght *
Math.sin(angle));
        int endY1 = (int) Math.round(centerPoint[1] + linesLenght *
Math.cos(angle));
        int endX2 = (int) Math.round(centerPoint[0] + linesLenght *
Math.sin(angle2));
        int endY2 = (int) Math.round(centerPoint[1] + linesLenght *
Math.cos(angle2));
        g.drawLine(centerPoint[0], centerPoint[1], endX1, endY1);
        g.drawLine(centerPoint[0], centerPoint[1], endX2, endY2);
    }
    g.setColor(Color.BLUE);
    g.drawLine(0, centerPoint[0], 100000, centerPoint[0]);
    try {
        ImageIO.write(bi, "PNG", new File(name));
    } catch (IOException e) {
        e.printStackTrace();
    }
}

}

class Graphic2 {
    static final long serialVersionUID = 0L;
    final double[] planetEccentricity = {0.20563593, 0.0068, 0.01671123, 0.0933941,
0.048775, 0.055723219, 0.044405586, 0.011214269, 0.2488273};
    final double[] planetPeriapsis = {29.124279, 54.85229, 114.20783, 286.46230,
275.066, 336.013862, 96.541318, 265.646853, 113.76329};
    final double[] planetPeridist = {0.30749951, 0.71843270, 0.98329134, 1.381,
4.950429, 9.048, 18.37551863, 29.76607095, 29.667};
    AffineTransform at = new AffineTransform();

    public void draw(double[] periapsis, double[] eccentricity, double[] peridist,
String name, double Multiplition) {

        BufferedImage bi = new BufferedImage(6000, 6000, BufferedImage.TYPE_INT_ARGB);
        Graphics2D g = bi.createGraphics();
        Graphics2D g2d = (Graphics2D) g;
        Font Font1 = new Font("TimesRoman", Font.BOLD, 70);
        g.setColor(Color.WHITE);
        g.fillRect(0, 0, 10000, 10000);
        g.setColor(Color.black);
        for (int i = 0; i < periapsis.length; i++) {
            drawEllips(g, periapsis[i], eccentricity[i], peridist[i] * Multiplition);
        }
        g.setColor(Color.YELLOW);
        g.fillOval(2990, 2990, 20, 20);
        g.setStroke(new BasicStroke(8.0f));
        g.setColor(Color.red);
        for (int i = 0; i < planetEccentricity.length; i++) {
            drawEllips(g, planetPeriapsis[i], planetEccentricity[i], planetPeridist[i]
* Multiplition);
        }
    }
}

```

```

        at.setToRotation(0, 3000, 3000);
        g2d.setTransform(at);

        g.setColor(Color.WHITE);
        g.fillRect(0, 5700, 10000, 300);
        g.setStroke(new BasicStroke(5.0f));
        g.setColor(Color.black);
        g.setFont(Font1);
        g.drawLine(3000, 5900, 6000, 5900);
        g.setStroke(new BasicStroke(8.0f));
        for (int i = 0; i < 100; i++) {
            double x1 = i * Multiplition + 3000;
            int x = (int) x1;
            g.drawLine(x, 5850, x, 5950);
            g.drawString(String.valueOf(i), x, 5800);
        }
        try {
            ImageIO.write(bi, "PNG", new File(name));
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public void drawEllipsis(Graphics g, double angle, double eccentricity, double peridist) {
        Graphics2D g2d = (Graphics2D) g;
        if (eccentricity < 1) {
            at.setToRotation(Math.toRadians(angle), 3000, 3000);
            g2d.setTransform(at);
            double a = peridist / (1 - eccentricity);
            double b = a * Math.sqrt(1 - Math.pow(eccentricity, 2));
            for (double t = 0; t < 2 * Math.PI; t += 0.0001) {
                double x = a - peridist + a * Math.cos(t);
                double y = b * Math.sin(t);
                drawPoint(g2d, x, y);
            }
        } else {
            at.setToRotation(Math.toRadians(angle + 180), 500, 500);
            g2d.setTransform(at);
            double a = peridist / (eccentricity - 1);
            double b = a * Math.sqrt(Math.pow(eccentricity, 2) - 1);
            for (double t = -1000; t < 1000; t += 0.001) {
                double x = a + peridist - a * Math.cosh(t);
                double y = b * Math.sinh(t);
                drawPoint(g2d, x, y);
            }
        }
    }

    void drawPoint(Graphics2D g, double x, double y) {
        int StartX = 3000;
        int StartY = 3000;
        g.drawLine(StartX + (int) x, StartY + (int) y, StartX + (int) x + 1, StartY +
(int) y + 1);
    }
}

class Graphic3{
    void drawGraphic(double[] eccentricity,double[] inclination,String name){
        int width = 10000;
        int height = 10000;
        if (eccentricity.length==inclination.length){
            BufferedImage bi = new BufferedImage(width, height, Buff-
eredImage.TYPE_INT_ARGB);

            Graphics2D g2d = bi.createGraphics();
            g2d.setColor(Color.white);
            g2d.fillRect(0, 0, width, height);
            g2d.setColor(Color.black);
            for(int i=0;i<eccentricity.length;i++){

```

```

        int x = (int)Math.round(eccentricity[i]*2000);
        int y = (int)Math.round(inclination[i]*100);
        /* if (inclination[i]<=10){
            y += 100;
        } else if (inclination[i]>=10&&inclination[i]<100){
            y +=110;
        }else if (inclination[i]>=100){
            y+=200;
        }*/

        g2d.fillOval(x+100, y+100, 14, 14);

    }
    /*double count=0;
    for(int i=0;i<height;i+=25){
        g2d.drawLine(80, 3500-i, 100, 3500-i);
        g2d.drawString(String.valueOf(count), 70, 9500-i);
        count+=0.5;

    }
    count =0;
    for(int i=0;i<4000;i+=50){
        g2d.drawLine(i+102, 9500, i+102, 9520);
        g2d.drawString(String.valueOf(count/10), i+90, 9540);
        count++;
    }
    g2d.drawLine(90, 0, 90, 9500);
    g2d.drawLine(100, 9510, 1000, 9510);

    /*g.setColor(Color.red);
    g.setStroke(new BasicStroke(8.0f));
    for (int i =0;i<eccentricity.length;i++){
        int x = (int)eccentricity[i]*1000;
        int y = (int) inclination[i]* 50;
        //drawPoint(g2d,x,y);
        g2d.fillOval(x, y,5,5);
    }*/
    try {
        ImageIO.write(bi, "PNG", new File(name));
    } catch (IOException e) {
        e.printStackTrace();
    }
}

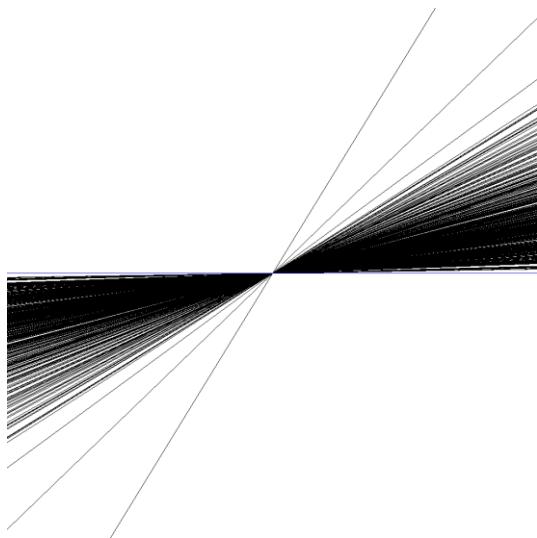
}

void drawPoint(Graphics2D g, double x, double y) {
    int StartX = 3000;
    int StartY = 3000;
    g.drawLine(StartX + (int) x, StartY + (int) y, StartX + (int) x + 1, StartY +
(int) y + 1);
}
}
}

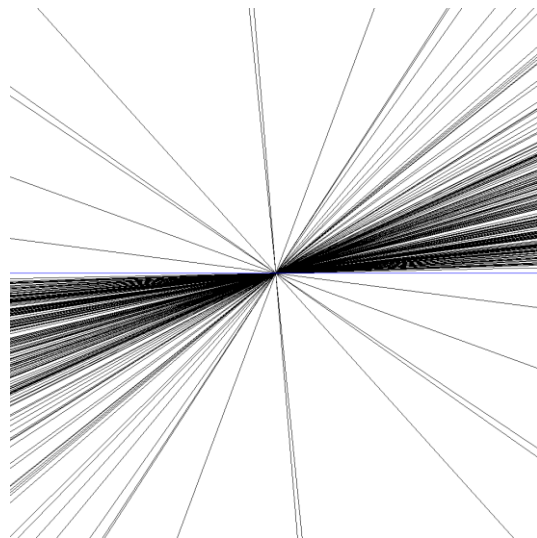
```

Наклонение орбит комет разных семейств

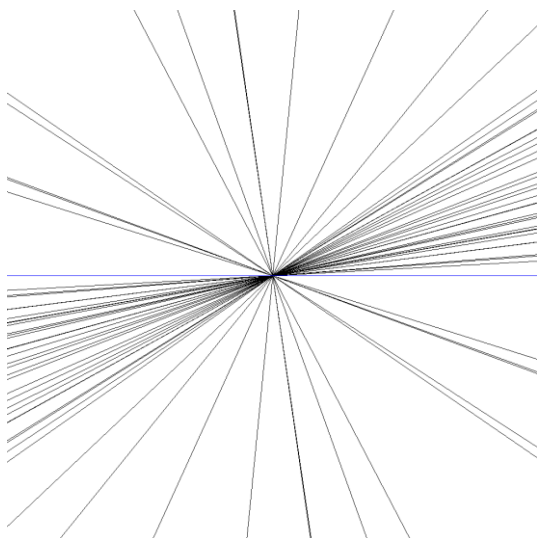
Семейство Юпитера



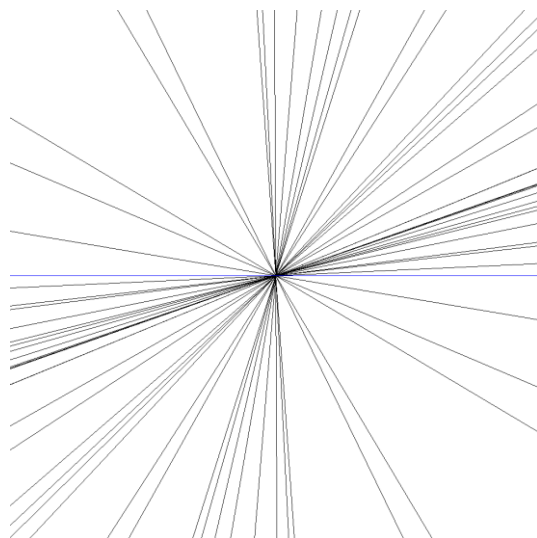
Семейство Сатурна



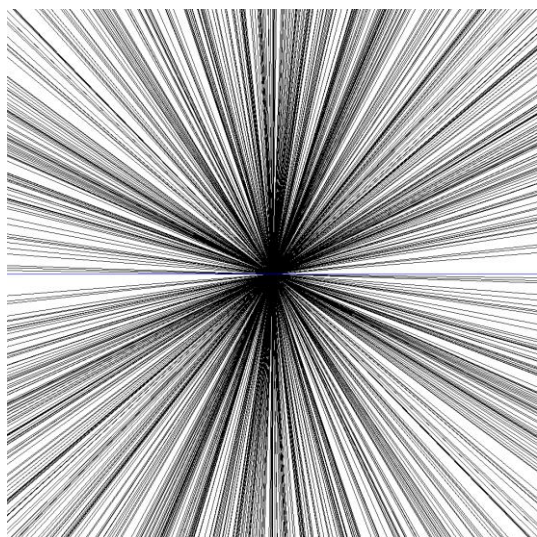
Семейство Урана



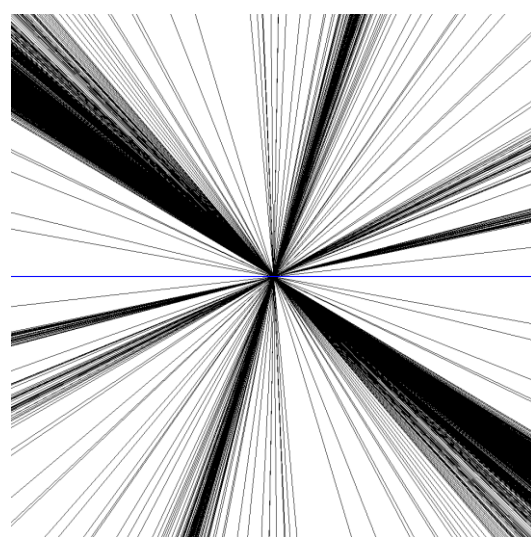
Семейство Нептуна



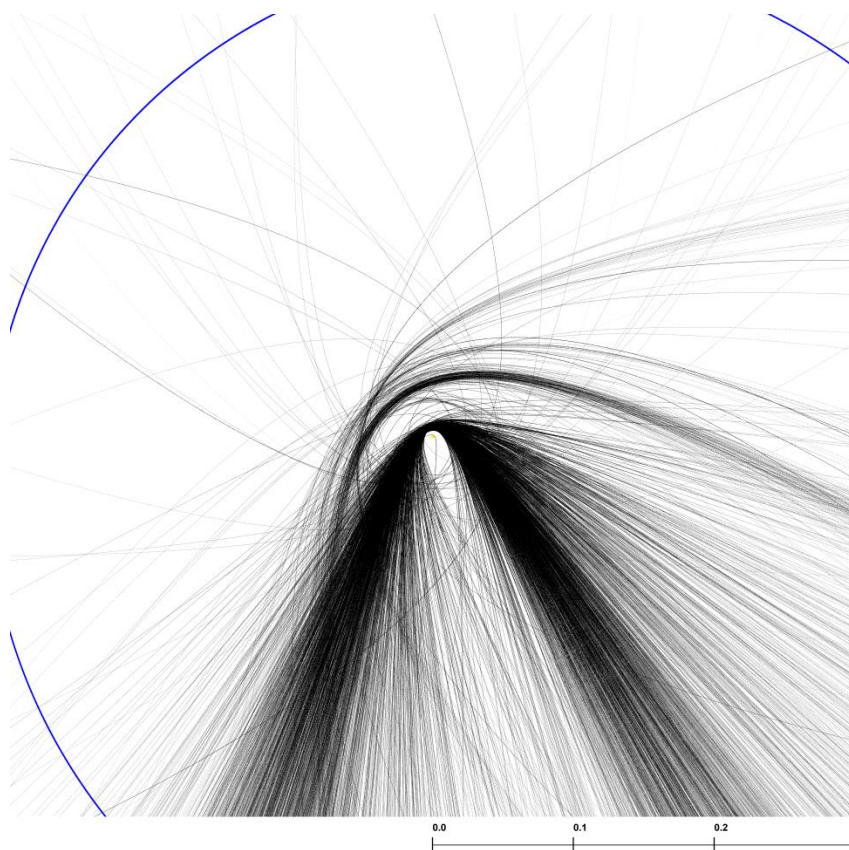
Семейство галлевского типа



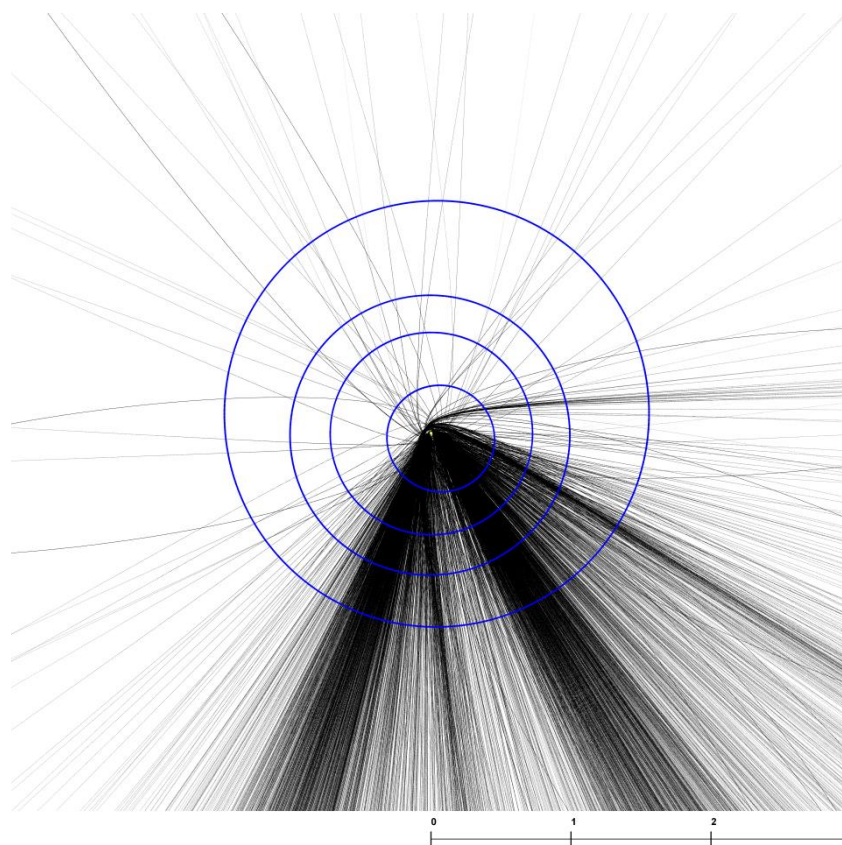
Семейство околосолнечных комет



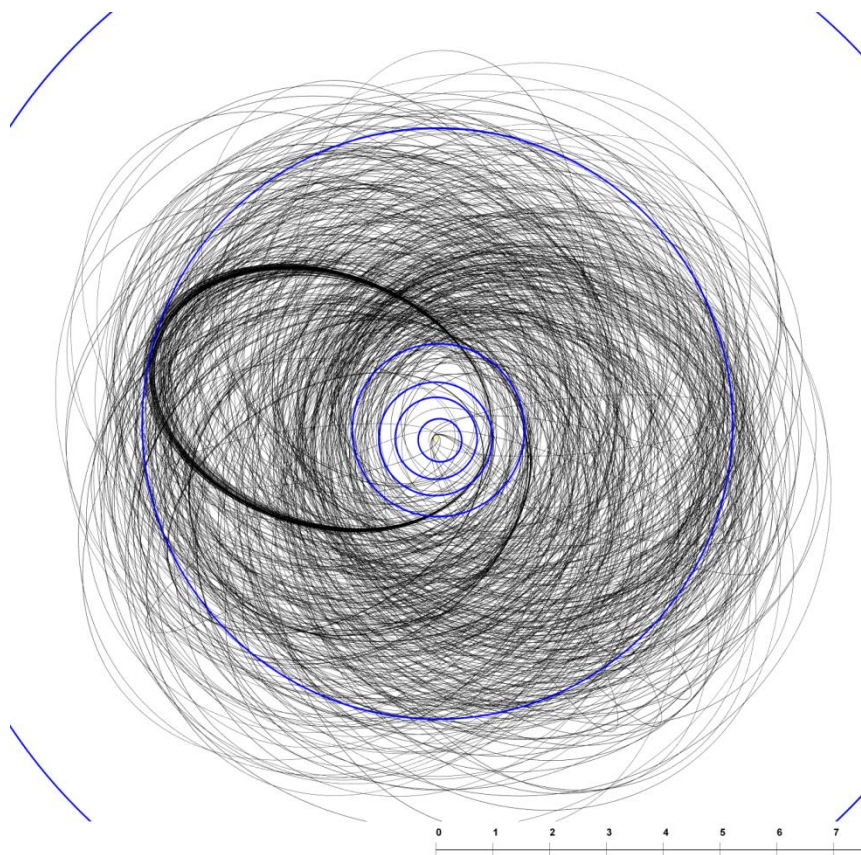
Положение орбит семейства околосолнечных комет



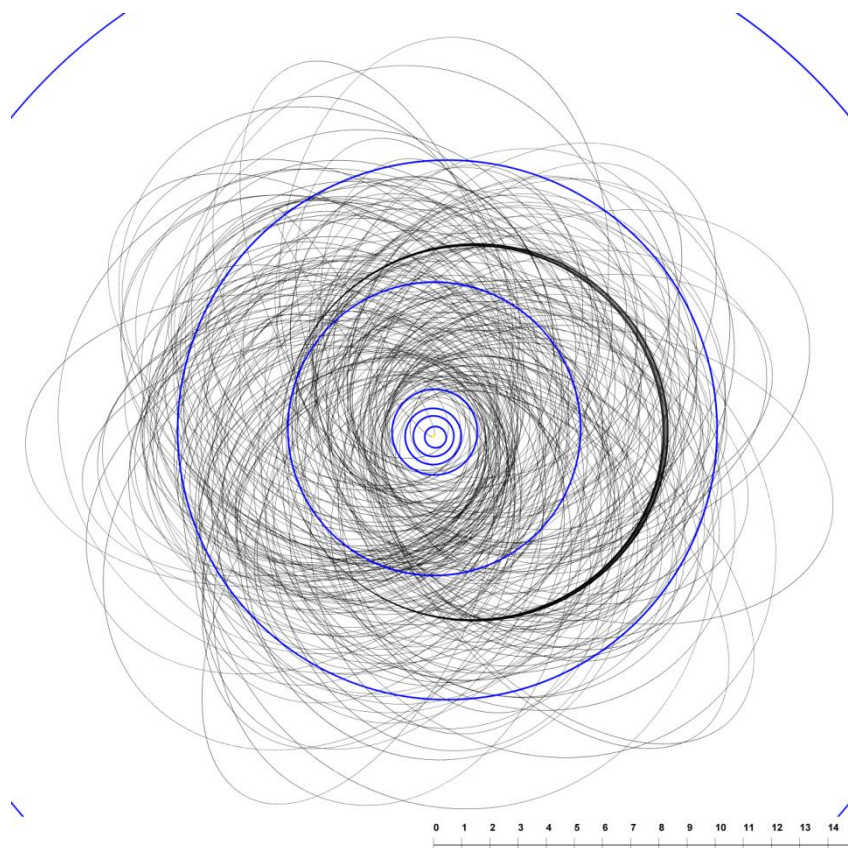
Положение орбит семейства околосолнечных комет



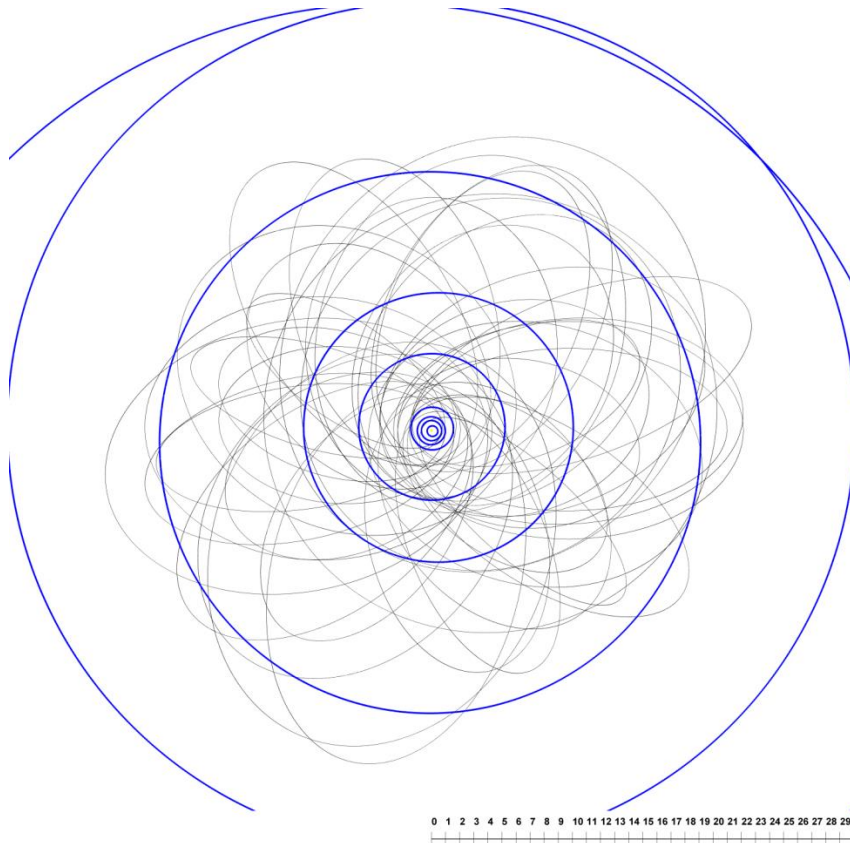
Положение орбит комет семейства Юпитера



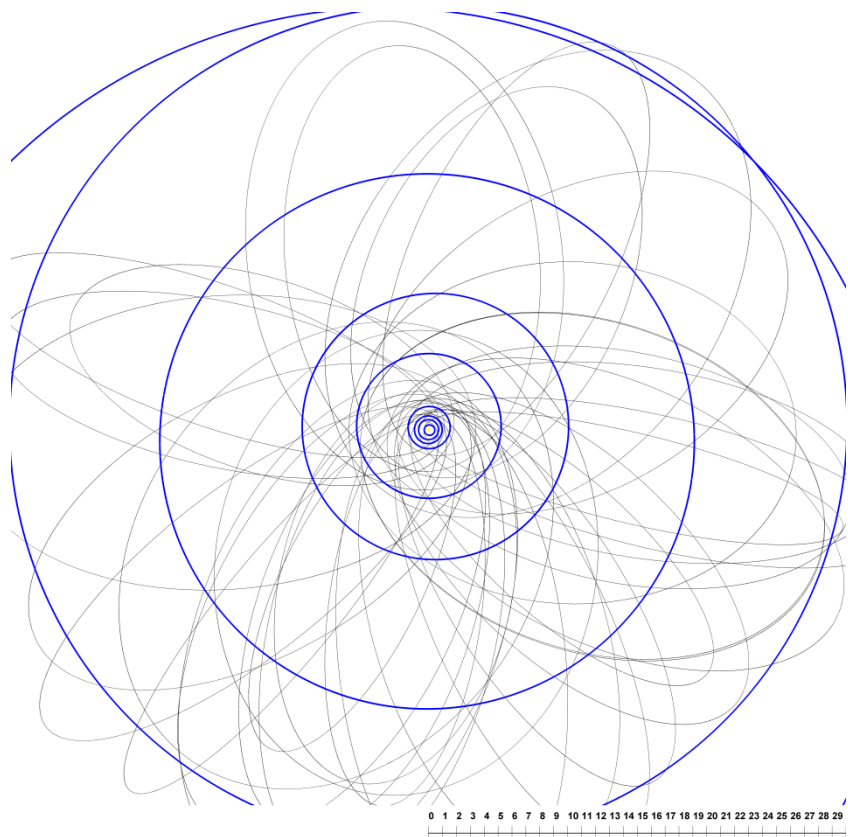
Положение орбит комет семейства Сатурна



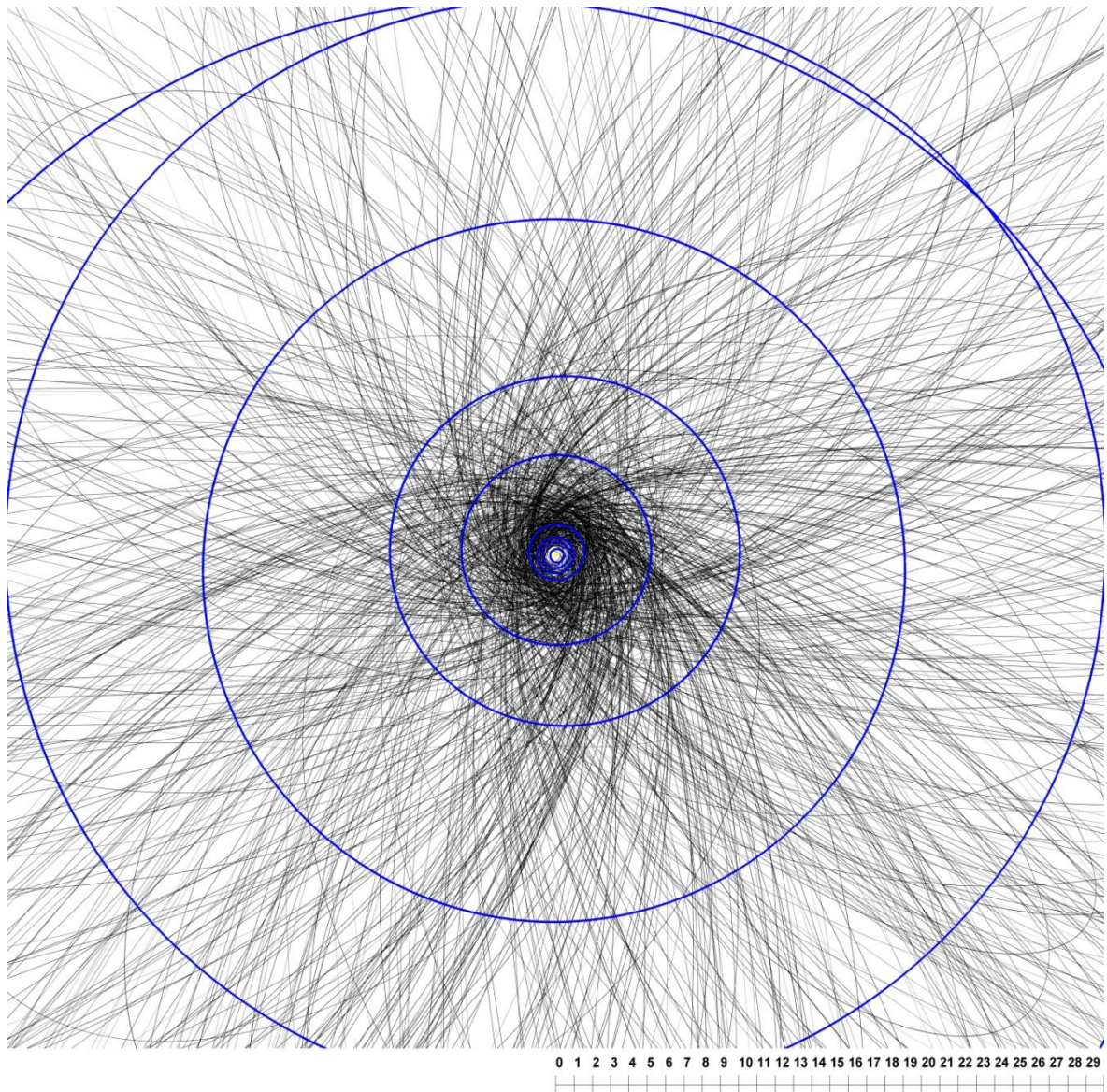
Положение орбит комет семейства Урана



Положение орбит комет семейства Нептуна



Положение орбит комет семейства галлеевского типа



Зависимость наклона орбит от их эксцентриситета

Ось абсцисс — эксцентриситет, ось ординат — наклонение. Синим цветом обозначены кометы семейства Юпитера, зелёным — Сатурна, жёлтым — Урана, розовым — Нептуна, красным — галлеевского типа.

