REPORT 2

Aayush Patil(22220)

1 Dataset Overview

The dataset comprises observations spanning from 1580 to 1604, with a 2-year interval and includes 15 features.

2 Mars Heliocentric Longitude Calculation

To compute the "Mars Heliocentric longitude" for each entry, we can use the following formula:

$$\text{Longitude} = \text{ZodiacIndex} \times 30 + \text{Degree} + \frac{\text{Minute}}{60} + \frac{\text{Second}}{3600}. \tag{1}$$

3 Time Difference Calculation

Next, we will determine the difference in days between two entries in the dataset. We will create a matrix called "oppositions" that will store both the "Mars Heliocentric longitude" values and the "time difference" values. This matrix will be instrumental in analyzing Mars's positions in its orbit.

4 Spokes Derivation

Using the data from the oppositions, we can derive spokes relative to the sun-Aries axis and the equant 0.

5 Global Function Definition

We will define a global function named `getIntersectionPoint(h, k, theta, r, c)`, where:

- h and k represent the coordinates of the intersection points;
- θ is the angle with the x-axis;
- r denotes the radius of the orbit;
- c indicates the angle at which the orbit's center is positioned concerning the sun.

6 Mars Equant Model

For Question 1, we will create a function called `MarsEquantModel()` with parameters:

- c: the orbit of the circle centered at angle c (in degrees);
- r: the radius of the orbit (measured in units of sun-center distance);
- $e1$ and $e2$: coordinates of the equant, where $e1$ is the distance from the sun and $e2$ is the angle in degrees relative to equant 0, which is defined by its angle concerning Aries;
- S represents Mars's angular velocity around the equant.

This function will return a list of errors for all 12 oppositions, along with the maximum error.

7 Parameter Optimization

7.1 Question 2

For Question 2, we will fix the values of r and s and perform a grid search over the parameters c, $e1$, $e2$, and z to minimize the angular error for a specific r and s. A function named `bestOrbitInnerParams(r, s, oppositions)` will be defined, which will output the parameters c, $e1$, $e2$, z, the error for 12 oppositions, and the minimum maximum error.

7.2 Question 3

For Question 3, we will fix the value of r and conduct a discretized search around 360 degrees over a period of 687 days. The function `bestS(r, oppositions)` will utilize `bestOrbitInnerParams()` for each value of s, providing the optimized value for s, the error for 12 oppositions, and the least maximum error.

7.3 Question 4

For Question 4, we will fix the value of s and perform a discretized search around the average distance of the black dots (the intersections of the dotted lines from the equant and the solid lines from the sun). A function called `bestR(s, oppositions)` will be defined, which will internally call `bestOrbitInnerParams()` for each value of r. This function will yield the optimized value for r, the error for 12 oppositions, and the least maximum error.

7.4 Question 5

For Question 5, we will define a function called `bestMarsOrbitParams()`, which will conduct a grid search in the vicinity of the best initial estimates for r and s. This function will internally call `bestOrbitInnerParams()` for each combination of r and s, ultimately providing the best parameter values for c, z, r, $e1$, $e2$, r, s, the error list for 12 oppositions, and the best maximum error.

8 Summary of Outputs

The following array contains the error values for the 12 spokes:

Errors = [0.06070340731442059, 0.004617946610977697, 0.015188522583429176, 0.0015322313151955314, 0.0268

The maximum error among these values is:

$$\text{Max Error} = 0.06773239519242225$$

The best parameters obtained from the function `bestMarsOrbitParams(oppositions)` are:

- $c = 149.0$
- $r = 8.599999999999998$
- $e1 = 1.6$
- $e2 = 93.19999999999999$
- $z = 55.80000000000001$
- $s = 0.5240937545494248$

9 Plots

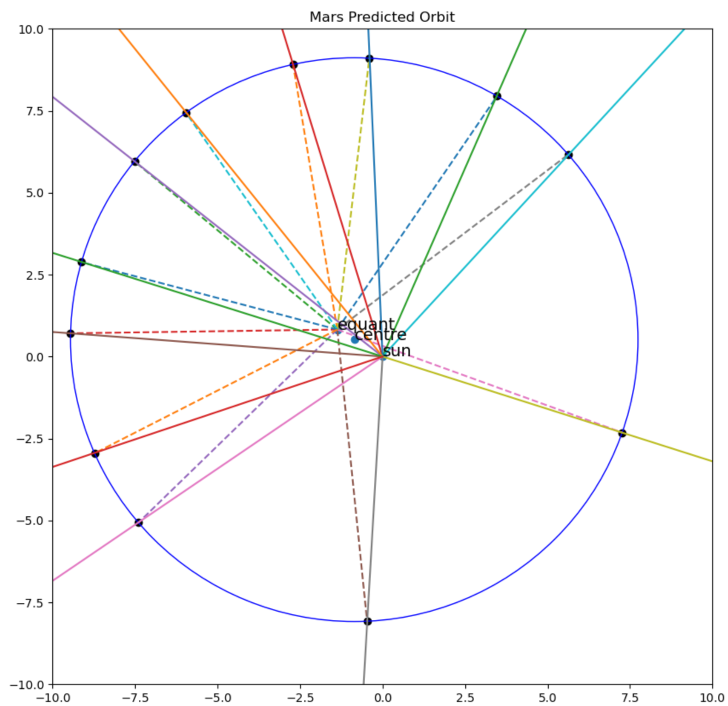

Figure 1: Plot 1: Predicted mars orbit