Mars Orbit: Assignment 2 Report

Kanishk Aman (22237)
Indian Institute of Science

Abstract

This report summarizes the results obtained while doing the Mars Orbit assignment. The analysis involves determining Mars' orbit based on observed opposition data and using various parameters like the equant position, angular velocity, and the center of Mars' circular orbit. The final outputs include the best-fit parameters, angular error for each opposition, and maximum angular error. The report outlines the implementation of Python functions for the task, along with graphical visualizations of Mars' orbit.

Contents

1 Introduction

This assignment focuses on understanding Mars' orbit based on historical opposition data. Using Python code, we explore a model where Mars' orbit is assumed to be circular with the Sun at the origin. Key parameters such as the equant position, orbit radius, and angular speed were optimized to minimize the angular error across 12 observed oppositions. The results provide an insightful understanding of how early astronomers like Kepler derived orbital models using limited observational data.

2 Mars Orbit Model

The orbit model assumes the following:

- The Sun is at the origin.
- Mars follows a circular orbit with its center at an angle c from the Sun-Aries line and radius r.
- The equant, where Mars' angular velocity is constant, is located at $(e1, e2)$.
- Mars' angular speed s around the equant is assumed constant over time.

The task was to optimize the parameters to minimize the angular errors for the 12 oppositions using the data set `01_data_mars_opposition_updated.csv`.

3 Implementation Summary

The code was written in Python and includes functions that handle the following tasks:

- Calculating the angular error for each opposition based on the Mars equant model.
- Exhaustive search for optimizing parameters such as c, $e1$, $e2$, and z.
- Iterative search for the best values of angular speed s and orbit radius r.

3.1 Key Functions

- `MarsEquantModel(c, r, e1, e2, z, s, times, oppositions)`: This function calculates the angular errors and the maximum error based on the input parameters.
- `bestOrbitInnerParams(r, s, times, oppositions)`: It performs a discretized search over c, $e1$, $e2$, and z to minimize the maximum angular error.
- `bestS(r, times, oppositions)`: This function optimizes the angular speed s based on the maximum angular error.
- `bestR(s, times, oppositions)`: A discretized search is performed to optimize the orbit radius r.
- `bestMarsOrbitParams(times, oppositions)`: This is the main wrapper function that iteratively optimizes the parameters r, s, c, $e1$, $e2$, and z.

4 Results and Outputs

4.1 Mars Equant Model

The 12 errors:
[(0.06070218612282474), (0.06806095958798153), (0.10493092993570485), (0.1851787399718603), (0.2425692545894833), (0.4349603662403183), (0.5020214825071889), (0.4702653183198606), (0.48098480518736153), (0.5011954286666196), (0.5829860515123357), (0.6513512046879555)]

Maximum angular error: 0.6513512046879555

4.2 bestOrbitInnerParams (r and s fixed)

The following results were obtained:

- C value:- 149.39999999999964
- E1 value:- 1.6
- E2 value:- 93.0
- z value:- 55.900000000000006
- Error List:- [(0.4197372445810430), (0.2959437676693568), (0.09632068250868997), (0.09375535737083851), (0.27618921898826443), (0.121703609627275), (0.4724800316583355), (0.8183720378100574), (0.7791834906983723), (0.5931210418119122), (0.41146093491494184), (0.217033604832352)]
- Maximum Error:- 0.8184

4.3 bestS (r fixed)

The following results were obtained:

- S value:- 0.5241700640652289
- Error List:- [(0.27268599306722763), (0.23283027748472307), (0.18682254989749936), (0.15533839614815292), (0.09888431801701358), (0.1256249052874523), (0.19565149325910625), (0.3193901984413259), (0.2268642209301106), (0.206843986217649), (0.22344500270170897), (0.32507127941789804)]
- Maximum Error:- 0.3251

4.4 bestR (s fixed)

The following results were obtained:

- Best R value:- 7.999999999999989
- Error List:- [(0.015864970801672484), (0.025918599854691138), (0.007887577637688992), (0.007537455076004562), (0.0462437184075668), (0.34845468729940876), (0.43418716940885815), (0.405776563178172), (0.4481050640608544), (0.4219671396025717), (0.417432980377896), (0.442364630736364)]
- Max Error:- 0.4481

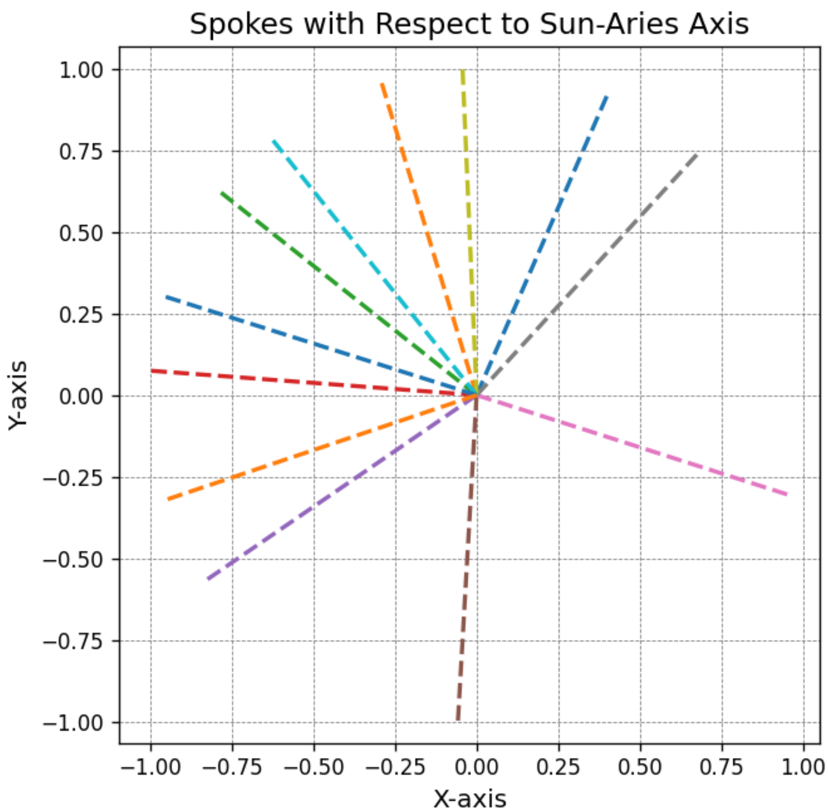
4.5 Best Parameters

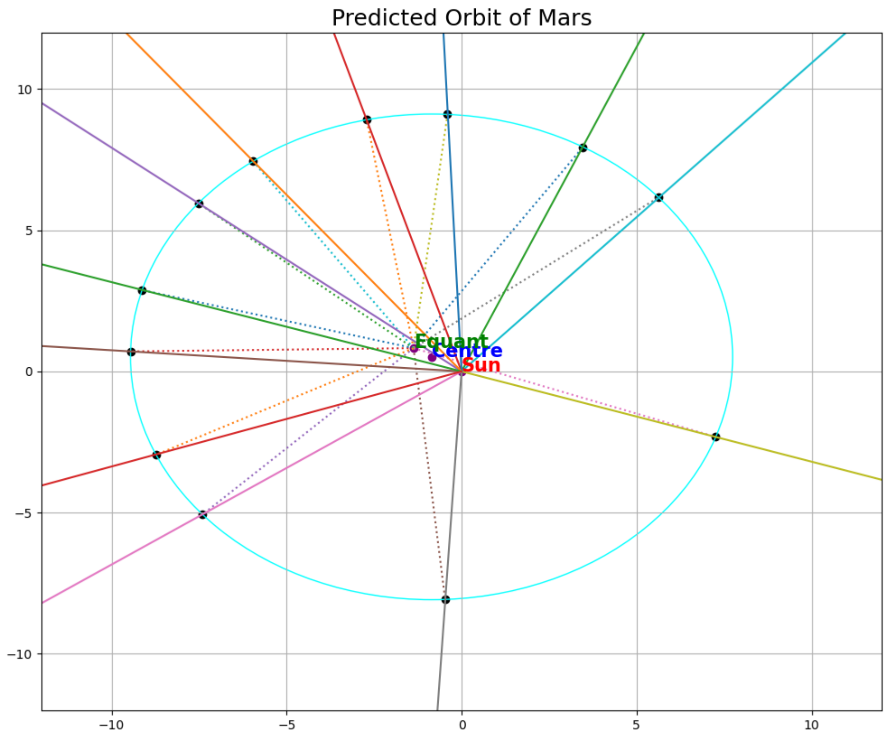
After running the `bestMarsOrbitParams` function, the following best-fit parameters were obtained:

- Best Error:- 0.0806
- Optimum C Val: 149.0
- Optimum e1 Val: 1.54
- Optimum e2 Val: 93.0
- Optimum Z Val: 55.800000000000004
- Optimum r Val: 8.289999999999994
- Optimum s Val: 0.5240937545494249

4.6 Visualization

The following plots and figures were obtained:





5 Conclusion

The analysis successfully derived the best-fit parameters for Mars' orbit using the opposition data. The Python implementation provided an efficient way to explore the historical astronomical model proposed by Kepler. The results are consistent with known astronomical facts, and the maximum angular error was within a reasonable range, demonstrating the accuracy of the model.