Mars Orbit: Assignment 1

This is going to be a tough first exercise. But in the end, you would have mastered the Mars Orbit module and would have also become very familiar with Python 3.9.0.

Rules

1. Use only Python 3.9.0. Any other version will cause difficulty in evaluation and will not be accepted.
2. You can discuss ideas with others. You can clarify model parameters and their descriptions. But don't discuss solutions or share code. The entire implementation from start to finish should be your own.
3. You will sign an ethics and integrity statement while submitting.
    a. If we detect a violation of the ethics and integrity requirement, we will immediately report it to the student's degree programme convenor and department chair. The grade for the assignment will be 0 for the first violation.
    b. If a second violation is detected, the convenor, department chair, and the Dean of your programme will be re-notified, and a Fail grade will be issued to the student.
4. How will we evaluate? We will examine your code manually, and will also use automated plagiarism checks. We will run your code on test inputs and assess how close they are to our reference implementation. We will run your code on the actual oppositions data, and will check whether the outputs match those in your Report1.pdf and whether they are close to the reference implementation's output.
5. **Deadline 23:59 hrs on Wednesday 25 August 2021. See the end of this assignment on what to submit on the Moodle portal. Remember to upload files, accept the submission statement, and click the submit button.**
6. Late submission policy:
    a. 0 hrs - 72 hours late: Only 50% credit
    b. 72+ hrs - 1 week late: Only 25% credit
    c. Beyond 1 week: No credit

The data set

01_data_mars_opposition_updated.csv (.csv file):

This file contains data on longitude/latitude of Mars under "opposition" with the Sun, in the ecliptic coordinate system.

a) Columns A/B/C/D/E are year/month/day/hour/minute of the opposition.
b) Columns F,G,H,I denote the ZodiacIndex, Degree, Minute, Second, respectively, of Mars's (heliocentric) longitude in the ecliptic coordinate system. ZodiacIndex refers to the zodiac (Aries 0, Taurus 1, ..., Pisces 11).
   Longitude = ZodiacIndex*30 + Degree + Minute/60 + Second/3600 (degrees)
c) Columns J,K refer to degree, minute of the geocentric latitudinal position of Mars in the ecliptic coordinate system.
d) Columns L,M,N,O refer to Mars's mean longitude, with reference to Kepler's approximated equant. Instead of using this, you will find these based on your own equant.

The Assignment 1

1. Assume the following orbit model and parameters:
    a. The Sun is at the origin.

b. The orbit is a circle with a specified centre at angle c (degrees), distance 1 from the Sun.
c. The orbit has radius r (in units of the Sun-centre distance).
d. The equant is located at (e1, e2) in polar coordinates with centre taken to be the Sun, where e1 is the distance from the Sun and e2 is the angle in degrees with respect to a particular reference longitude (called 'equant 0').
e. The reference longitude (equant 0) is given by angle z (degrees) with respect to Aries.
f. The angular velocity of Mars around the equant is s degrees per day.

Write a function to take these as input in the above order, along with the twelve oppositions, and return the following: angular error for each opposition and the maximum angular error:

errors,maxError = MarsEquantModel(c,r,e1,e2,z,s,oppositions)

The two variables, oppositions (in degrees and in decimal) and errors (degrees, decimal), are arrays; all others are scalars.

[Internally, you can structure your code in a suitable way with your own useful functions.]

2. Fix r and s. Do a discretised exhaustive search over c, over e = (e1,e2), and over z to minimise the maximum angular error for the given r and s. Your outputs should be the best parameters, the angular error for each opposition, and the maximum angular error, as follows.
c,e1,e2,z,errors,maxError = bestOrbitInnerParams(r,s,oppositions)

3. Fix r. Do a discretised search for s (in the neighbourhood of 360 degrees over 687 days; for each s, you will use the function developed in question 2). Your outputs should be the best s, the angular error for each opposition, and the maximum angular error, as follows.
s,errors,maxError = bestS(r,oppositions)

4. Fix s. Do a discretised search for r (in the neighbourhood of the average distance of the black dots, which are described in slide 31, from the centre; again, for each r, you will use the function developed in question 2). Your outputs should be the best r, the angular error for each opposition, and the maximum angular error.
r,errors,maxError = bestR(s,oppositions)

5. Write a wrapper that will search iteratively over r and s, starting from an initial guess. Your outputs should be the best parameters c, e, z, r, and s, the angular error for each opposition, and the maximum angular error.
r,s,c,e1,e2,z,errors,maxError = bestMarsOrbitParams(oppositions)

What should you submit?

You should submit your Python 3.9.0 code and a report as a single zip file, named as **YourName_LastFiveDigitsOfYourSRNumber_Assignment1.zip**, on Moodle before the deadline, and this zip file should contain the following:
Assignment1.py
Report1.pdf
Report1.pdf should contain your name as the author, the outputs that you obtained when you ran **bestMarsOrbitParams(oppositions)**, and an implementation summary.