Mars Orbit

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.

1. Assuming the following orbit model and parameters:
   a. The Sun is at the origin.
   b. Mars's orbit is circular, with the centre at a distance 1 unit from the Sun and at an angle c (degrees) from the Sun-Aries reference line.
   c. Mars's 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 the Sun-Aries reference line.
   e. The 'equant 0' angle z (degrees) which is taken as the earliest opposition, also taken as the reference time zero, with respect to the equant-Aries line (a line parallel to the Sun-Aries line since Aries is at infinity).
   f. The angular velocity of Mars around the equant is s degrees per day.

   It will be helpful to draw a picture with the variables.

   Write a function to take these as input in the above order, along with the twelve times and 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,times,oppositions)

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

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,times,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 part 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,times,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,times,oppositions)