**The data set**

Mars\_Opposition\_data.csv (.csv file):

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

ecliptic coordinate system.

1. Columns A/B/C/D/E are year/month/day/hour/minute of the opposition.

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)

1. Columns J,K refer to degree, minute of the geocentric latitudinal position of Mars in the ecliptic coordinate system.
2. 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.

**Assumptions:**

* 1. The Sun is at the origin.
  2. 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.
  3. Mars’s orbit has radius r (in units of the Sun-centre distance).
  4. 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.
  5. 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).
  6. The angular velocity of Mars around the equant is s degrees per day.

1. **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.

1. Fixed r and s. Performed discretised exhaustive search over c, over e = (e1,e2), and over z to minimise the maximum angular error for the given r and s. Outputs are 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)**

1. Fixed r. Did a discretised search for s (in the neighbourhood of 360 degrees over 687 days; for each s.

**s,errors,maxError = bestS(r,times,oppositions)**

1. Fixed s. Did a discretised search for r.

(**r,errors,maxError = bestR(s,times,oppositions)**