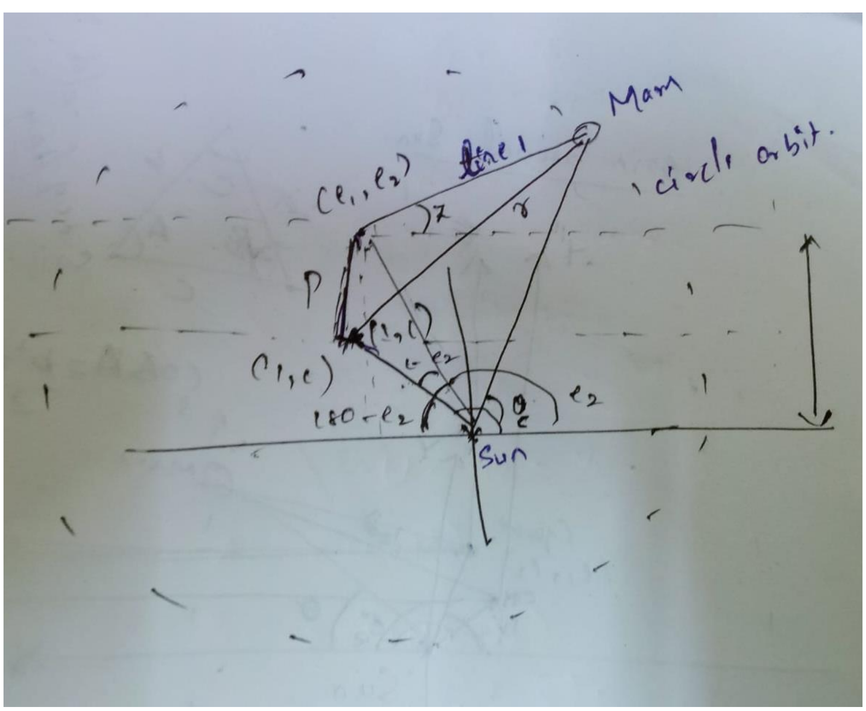
ASSIGNMENT 2

Mars Orbit

Assumptions:

- The Sun is at the origin.
- 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.
- Mars's orbit has radius r (in units of the Sun-centre distance).
- 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.
- 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).

Derivation of the Objective function



$$\text{Centre } (x_c, y_c) = (\cos c, \sin c)$$

$$(x - \cos c)^2 + (y - \sin c)^2 = r^2 \longrightarrow \text{Eqn of Circle orbit.}$$

$$y = mx + c \qquad m = \tan z \longrightarrow \text{Equation of Segment line}$$

$$e_y = \tan z (e_x) + c \Rightarrow c = e_y - e_x \tan z.$$

$$y = \tan z (x) + e_y - e_x \tan z. \longrightarrow \text{Equation of line}$$

$$(x - \cos c)^2 + (e_y + \tan z (x) - e_x \tan z - \sin c)^2 = r^2 \rightarrow \text{Circle eqn.}$$

$$(x - \cos c)^2 + (x(\tan z) - k)^2 = r^2$$

$$x^2 + \cos^2 c - 2x \cos c + x^2 \tan^2 z + k^2 - 2kx \tan z = r^2.$$

$$x^2 \underbrace{(1 + \tan^2 z)}_{A} - 2x \underbrace{(\cos c + k \tan z)}_{B} + \underbrace{\cos^2 c + k^2 - r^2}_{C} = 0$$

$$x = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A}$$

$$B = -2(\cos c + (e_y - e_x \tan z - \sin c) \tan z)$$

$$x_1 = \frac{-B + \sqrt{B^2 - 4AC}}{2A} \qquad x_2 = \frac{-B - \sqrt{B^2 - 4AC}}{2A}$$

$$y_1 = (x_1) \tan z + e_y - e_x \tan z.$$

The first question objective function is done as shown in the above figure.

The function returns two theta values out of which the one nearest to the given heliocentric longitude is taken for optimizing the values of the parameters.

The loss function is defined as the absolute difference between the actual heliocentric longitude and the predicted longitude from the model

While using trail and error method the optimal values of the parameters were found to be

[c= 2.599, r = 8.18, e1 = 1.5, e2 = 2.5, z = 0.974, s = 0.009144] all the angle values are in radians and the error given by model is

```
ERR0R = [0.0008045685106594647,
  0.000814279251742045,
  0.0002539088049546834,
  0.00012905960575837483,
  0.0008256697365807142,
  0.0008251489293931158,
  0.0008256072930485914,
  0.0008086725399367722,
  0.0007248692134043111,
  0.0008258635815399096,
  0.0006619172159125775,
  0.00039780194820115966]
```

Max Error = 0.0008258635815399096

The values above are in radians and if we convert that into minutes the ma ximum error is **2.8391 minutes**

2. From the first question we found out the optimal values of s and r and the error is very sensitive to the values of s and r, the values of s and r that was used are
S = 0.524
R = 9

The Scipy minimize was used to determine the parameters and the error and the result obtained are

C = 87.98401
e_1 = 1.51405
e_2 = 150.769
Z = 57.072

```
ERROR = [0.011194042009446958,
  0.006976599918616877,
  0.02047058741780905,
  0.02013827553580816,
```

```
  0.015644341865240996,
  0.020470584100531042,
  0.005447228353649081,
  0.020470585550390608,
  0.008164038082495884,
  0.009488976032684615,
  0.014113351347333047,
  0.009321084639543287]
```

MAX ERROR = 0.02047058741780905

The error and the max error values are in radians now if we convert that int o minutes the maximum error is 70.37269579164386 **minutes**

3. The objective function was used to determine the optimal values but thi s as to be done for a fixed value of r and from the value of r obtained in the first problem, we know that it should be somewhere between 8 and 10

So we are fixing r = 9

And the values of the parameters obtained are as follows

S = 0.524079
C = 148.186
e_1 = 1.66704
e_2 = 148.673
Z = 55.8620

```
ERROR = [1.01342864e-03, 2.10552453e-04, 8.29232580e-05, 4.31074038e-04,
        7.67685084e-04, 1.13957659e-03, 1.11868500e-03, 2.26871090e-05,
        4.31890965e-04, 5.11827539e-04, 1.13954926e-03, 2.27178463e-04]
```

MAX ERROR = 0.0011395765858326357

The error and the max error values are in radians now if we convert that int o minutes the maximum error is 3.91757 minutes

4. The objective function was used to determine the optimal values but thi r as to be done for a fixed value of s and from the value of s obtained in the first problem, we know that it should be somewhere between 0.52 and 0. 53

So we are fixing r = 0.524

And the values of the parameters obtained are as follows

r = 8.20408

```
ERROR = 0.00378354, 0.00346768, 0.00369794, 0.0037239 , 0.00347491,
        0.00114336, 0.00363851, 0.00313093, 0.00378354, 0.00367887,
        0.00378173, 0.0037782
```

MAX ERROR = 0.003783544798805938

The error and the max error values are in radians now if we convert that int o minutes the maximum error is 13.00686 minutes

5. The main problem is solved by using two nested for loops where one runs for r and the other runs for s in grid size of 20 and 120 respectively
The inner loop runs the optimize.min function to narrow down to the optimal values and after both the for loops is being run the values which as the min total error is taken and the values corresponding to that values the min error
The obtained values are

r = 8.5263, s = 0.5241, c = 151.4111, e1 = 1.5844, e2 = 149.0312, z = 55.7740

Errors = [8.25539672e-04 2.77899376e-04 1.81484975e-04 1.60193849e-04
 8.13945009e-04 8.26858722e-04 7.26870298e-04 8.23453443e-04
 4.07611965e-04 4.38131760e-05 3.04946105e-04 8.26858723e-04]

The maximum angular error = 0.0008

The error and the max error values are in radians now if we convert that into minutes the maximum error is **2.75019 minutes**