Report: Mars Orbit

Q1. Errors ,maxError = MarsEquantModel (c,r,e1,e2,z,s,oppositions)

Logic:

Solving Equation

Where D: Dotline Angle from Aries (e2 +z)

(Cx,Cy): Centre of Orbit Cartesian Coordinate (C)

(Ex,Ey): Equant Cartesian Coordinate from (E1,E2)

$(X - Cx)^2 + (X - Cy)^2 = R^2$

$Y - Ey = (X - Ex) * \tan D$

simplifying above equation to 2nd Order Equation and solve it

$$(1 + (\tan D)^2)\ X^2 + (-2Cx + 2\tan D\ (Ey - Cy - Ex * \tan D)X + (Ey - Cy - Ex * \tan D)^2 + Cx^2 + R^2 = 0$$

From Which we get (x1,y1), (x2,y2) by Discarding one point by checking Dotline (Stroke angle) lies in which XY quadrant (First and Fourth Quadrant X1 >0 and opposite in Third and Second)

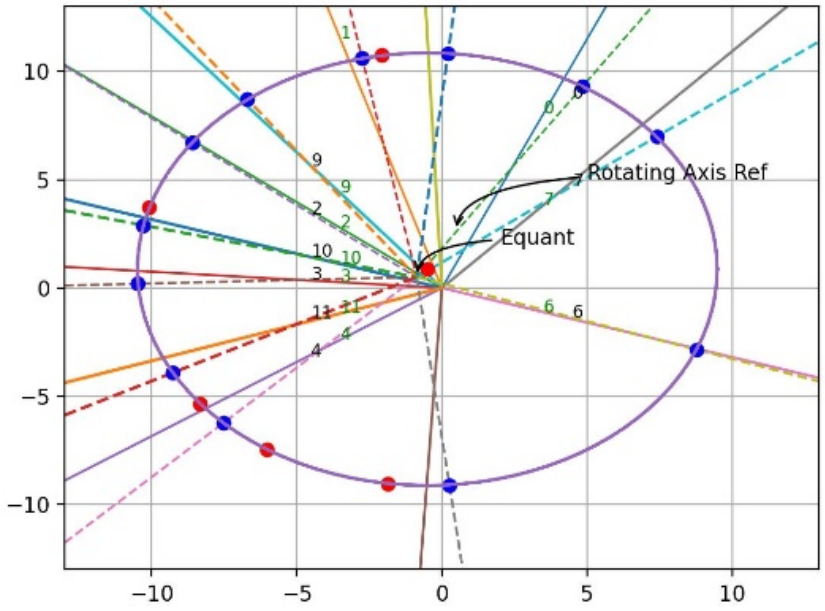
Return Value:

Error =Calculated Angle- Actual Longitude Angle

Max Error= Max of absolute Error

Result:

```
c= 120
e1= 1
e2= 93
z= 57
r= 10
s= 0.5240174672489083
T= 687.0
Error= [-4.01940046 -2.40252629  0.42688867  3.24152174  5.25834727  4.93203263
 -0.2582402  -4.1830855  -3.59949894 -1.06671242  1.84417968  4.28423937]
MaxError= 5.258347271638655
```



Q2:c,e1,e2,z,errors,maxError = bestOrbitInnerParams(r,s,oppositions)

Logic:

1) Approx. Calculation:
   x0i (Initial Approx. Functional Parameter Variable)
   Logic:(Grid Search Method One Parameter Varied Over range by Keeping fix remain at time)
2) Fine Control Calculation:
   - Optimized Method Used by SciPy Method : Nelder-Mead
   - LoopValue= Control the Flow of Search Condition LoopValue Remains True If Max Error of Function is (Error>= 100,10<= Error<=100,0.3<= Error<=10) correspond incremental Initial parameter given to optimize function By Method 1,2,3 Respectively

Method 1:

If Max Error is >=100 than Initial Value Start from X0i with Inc in Parameter Increment is Define by Zf -(0,360) if its Exhaust than Next (Cf, E2f) -((0,360),(0,360)) Than E2 Varies (1,.5*rf)

Method 2:

If Max Error range (10,100) Increment (Step .3 of Last Value Optimize Function Calculated) is Define by changing Cf & E2f to check the conversion is finite

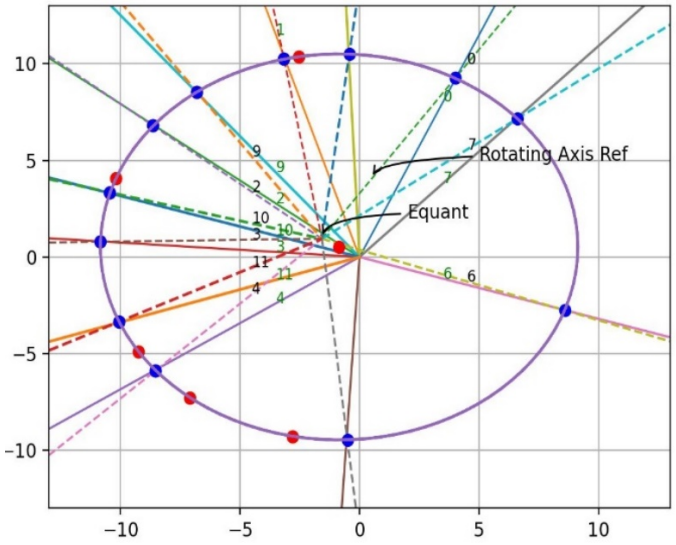
Method 3:

If Max Error range (1,10)
Passing Same Optimize Value to Optimize function And Checking
Divergence Value(fun(Second Last Max Error)-result.fun(Last Max Error))

Note: Total Fine Search Iteration Limit to 30

Result:

```
c= 148.466136528117
e1= 1.8371136185466113
e2= 92.81140799988856
z= 56.13007086821349
r= 10
s= 0.5240174672489083
T= 687.0
Error= [ 0.12884425  0.20685473  0.18592579  0.11816256  0.18011789  0.2001806
 -0.06349447 -0.20705674 -0.14000267 -0.12526152 -0.20705683 -0.20353295]
MaxError= 0.20705682624765132
```



Q3. s,errors,maxError = bestS(r,oppositions)

Logic:

Searching minimum Max error in range of Si in step of Precision_Control Factor(Default:30) deciding New Range Over

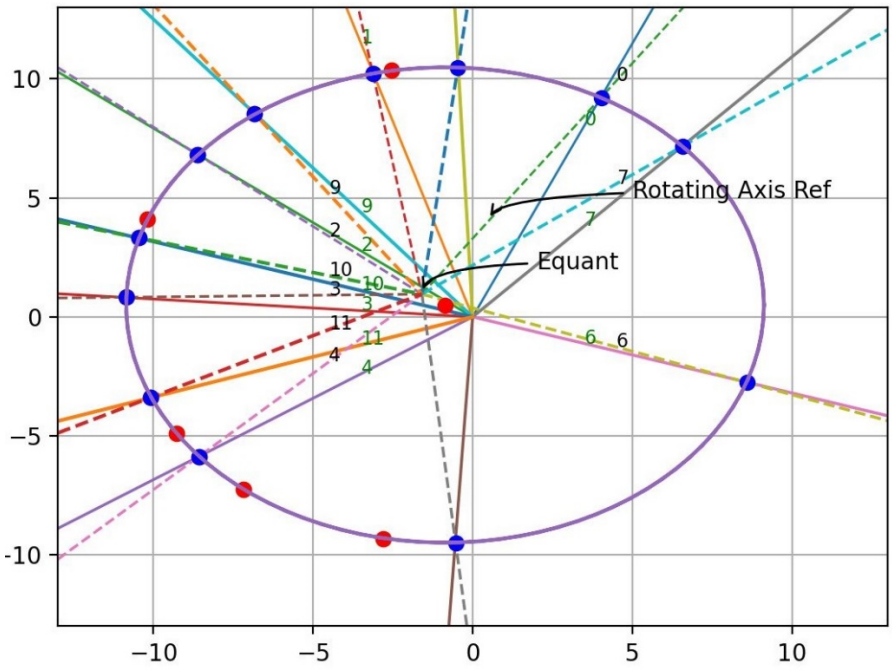
New range =Si(Minerror Index to -1, Minerror Index to +1)

Loop 1 Range :(S1,S2) ==> Mer
Loop 2 Range :(S1',S2') ==> Mer'
Loop 3 Range :(S1'',S2'') ==?Mer''

Result:

Using initial radius = 10 below new best S obtained and by using Q.2. Best Orbit Inner parameter obtained which result showed as below:

```
c= 150.01078309311606
e1= 1.854866301256326
e2= 93.17835760249761
z= 55.809242189981845
r= 10
s= 0.524084300416541
T= 686.9123912200248
Error= [-0.06085968  0.06085744  0.02617061 -0.06085968  0.00290761  0.06085968
 -0.06085959 -0.04039737  0.05990561  0.04807908 -0.05769798 -0.03955006]
MaxError= 0.06085967587466712
```



Q4: r,errors,maxError = bestR(s,oppositions)

Calculation Rf:

1) Solving Equation Y=X *tan(Act_Line) and (Y-ey)=(X-ex) *tan(Stroke)
   Simplified Equation X=(ey - ex * tan(Stroke)) / (tan(Act_Line) - tan(Stroke)
2) Calculate Each (X,Y) distance from (Cx,Cy)
3) New_Rf= is mean of Each distance between (X,Y) and (Cx,Cy)

Search Control:

calculate New_rf until the Error Start increasing continuously

Implement flow:

1)Appending Each iteration Value
  i) R_New (In r_Vari)
  ii) Max Error (In Mer)
  iii) Error_Change (Append Value True if error_tol became positive else false)
  Where: Error_tol: it describe error change from last value

2)Checking Divergence Condition:(Success than Error=True)
  i) If R_New decreasing ==> Break Internal Loop & Increment Factor += (Default :.2)& Set Error True
  ii) If R_New Exponentially Increasing ==>Break Internal Loop & Increment Factor -= Set Error True
  iii) Above Condition Fail Than Set Error False to deactivate Outer Loop

3) Check if more than (LoopValue/2) (Default 25) no of times error_tol become True Inner While loop breaks
  (Note : error_tol positive means --error curve(Initial Decreasing ,Makes Min, started increasing) in last part of started raising)

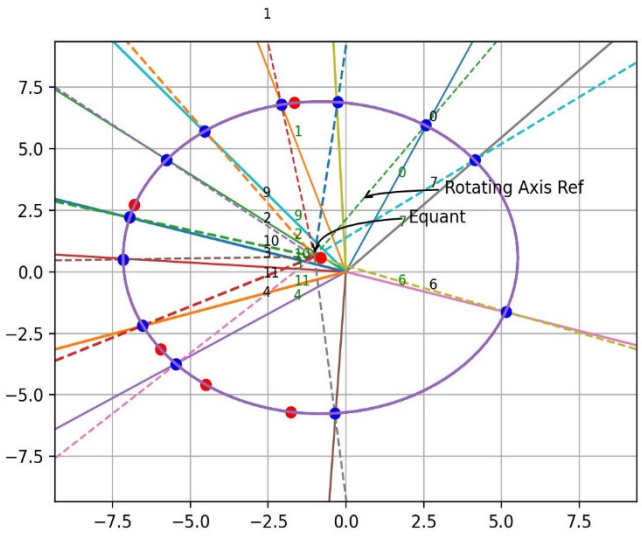
4) Final Rf correspond to min Max Error (Find by Mer,r_Vari)

5) Outer Loop Control by Error Value + (Max Search range of Rf(Default 20)) + (LoopValue= Max No. time innerloop Diverge)

Result:

Using initial S= 360 / 687 below new best R with default Initial Guess (R0=5) obtained and by using Q.2. Best Orbit Inner parameter obtained which result showed as below:

```
c= 144.85347645933558
e1= 1.1861591570252594
e2= 92.49679493387467
z= 56.19187326335065
r= 6.350418762366428
s= 0.5240174672489083
T= 687.0
Error= [ 0.25221158  0.04159575  0.14861747  0.25221163  0.13308176 -0.2520
  0.24745902  0.13164763 -0.25221164 -0.24303986 -0.11663796 -0.12287224]
MaxError= 0.25221163908881294
```



Q5. r,s,c,e1,e2,z,errors,maxError = bestMarsOrbitParams(oppositions)

Logic:

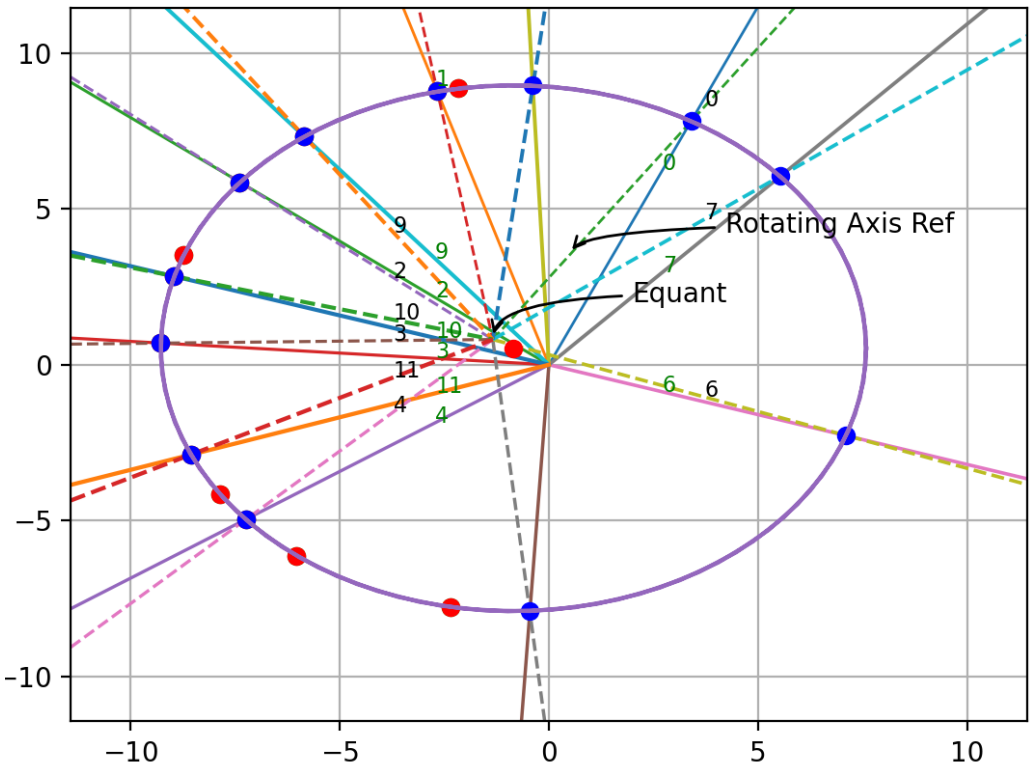
Simultaneously running the BestR and BestS function with initial Guess rf(Default : 10) and sf(Default: 360/687) until constrain gets true

Constrain:

1) Max Error < 4 Min. (Implemented using while condition)
2) if Max Error change diverge in last 2 complete iteration in way
    i) Remain Constant
    ii) After each BestR and BestS use Max error get increased from its last iteration
        (Implemented by tracing Error_Change last 4 Parameter)

Result:

```
c= 148.03829746498033
e1= 1.5679189018083783
e2= 93.0151312351963
z= 55.860515296494555
r= 8.44020401298252
s= 0.5240780728672885
T= 686.9205537075815
Error= [-0.03647562 -0.00735401  0.0197923   0.01337384  0.03630267 -0.03647581
  0.03637543  0.02212279 -0.02792524 -0.02328557 -0.03647343 -0.0029549 ]
MaxError= 0.036475812114531436
```



Conclusion:

Model is Depending too much on Orbit radius and the Angular Speed (S) below two result showing error can be reduce below 4 min.(.0667 deg) with two different.

```
c= 150.01078309311606
e1= 1.854866301256326
e2= 93.17835760249761
z= 55.809242189981845
r= 10
s= 0.524084300416541
T= 686.9123912200248
Error= [-0.06085968  0.06085744  0.02617061 -0.06085968  0.00290761  0.06085968
 -0.06085959 -0.04039737  0.05990561  0.04807908 -0.05769798 -0.03955006]
MaxError= 0.06085967587466712
```

Result 1 by keeping Orbit radius Fixed (r=10)

```
c= 148.03829746498033
e1= 1.5679189018083783
e2= 93.0151312351963
z= 55.860515296494555
r= 8.44020401298252
s= 0.5240780728672885
T= 686.9205537075815
Error= [-0.03647562 -0.00735401  0.0197923   0.01337384  0.03630267 -0.03647581
  0.03637543  0.02212279 -0.02792524 -0.02328557 -0.03647343 -0.0029549 ]
MaxError= 0.036475812114531436
```

Result 2 by keeping Orbit radius (r) and Angular Speed (s) simultaneously varying

Observation:-

Additional Detail Required to Verify Model Detail as for Two different Mars Orbit Parameter(With too much difference between radius(1.56 AU)) Period giving within tolerance limit.