```matlab
function root = secant(f,x1,x2,n)
    iteration=0;
    x(1)=x1;
    x(2)=x2;
    for i=3:1000
        x(i) = x(i-1) - (f(x(i-1)))*((x(i-1) - x(i-2))/(f(x(i-1)) - f(x(i-
2))));
        iteration=iteration+1;
        if abs((x(i)-x(i-1))/x(i))*100<n
            root=x(i);
            iteration=iteration;
            break
        end
    end
end
syms t;
xy1(t)= -11.9084+57.9117*cos(2*pi*t/87.97)*(-149.5832*sin(2*pi*t/365.25));
xy2(t)= -2.4987+149.6041*cos(2*pi*t/365.25)*(56.6741*sin(2*pi*t/87.97));
xM2(t)= (-2.4987+149.6041*cos(2*pi*t/365.25))^2;
yM2(t)= (56.6741*sin(2*pi*t/87.97))^2;
xE2(t)= -2.4987+149.6041*cos(2*pi*t/365.25)^2;
yE2(t)= -149.5832*sin(2*pi*t/365.25)^2;
f(t) = xy1(t)-xy2(t)/sqrt(xM2(t)+yM2(t))* sqrt(xE2(t)+yE2(t));
g(t) = diff(f);
 x1=input('Enter first point of guess interval: ');
 x2=input('Enter second point of guess interval: ');
 n=input('Enter allowed Error in calculation: ');
for i=1:10

    root = secant(g,x1,x2,n);
    x1 += 0.1;
    x2 += 0.1;
    disp(root)
end
```

I wrote a secant method which calculates the root of non-linear equation. It takes a function and x1 and x2 which show the interval that we assume the root is in that interval. The fourth input is the tolerance which show how much error can our function have.We have our function as follows:

$$f = \frac{x_M \times y_E - x_E \times y_m}{\sqrt{{x_M}^2 + {y_M}^2} \times \sqrt{{x_E}^2 + {y_E}^2}}$$

For obtaining the opposite position of planets we should consider the maximum distance. So we define function g as a differential of function f based on time t.