```
%Gustavo Grinsteins
%ASEN 5050
%HW6
%Solving lambert's equation

function aCalc = EllOrbitLambertEqSolve(s,c,a_m,TOF,TOFmin,mu,less180)
    %STEP 5: Usin Fsolve
    %Initial Guess for a
    a_initial = a_m+150;%how to defend delta_a?
    %Calc alpha boolean
    if TOF > TOFmin
        greatThanTOFmin = true;
    else
        greatThanTOFmin = false;
    end
    diff = 1000; %initial value for stopping condition
    diff2 = inf; %initial value for second stopping condition
    a_new = 0;
    iterations = 0;
    tolerance = 10^-5; %5 digit accuracy is desired
    while diff > tolerance %Convergence stopping condition
        %implement Fsolve function
        %define the anonymous function handle
        options = optimoptions('fsolve','Display','off');
        a_bef = a_new;
        a_new = fsolve(@(a)LambertEqt(mu,a,s,c,TOF,less180,greatThanTOFmin),a_initial,↙
options);
        %Recalculate values
        n_new = sqrt(mu/(a_new^3));
        alpha_0 = 2*asin(sqrt((s)/(2*a_new)));
        beta_0 = 2*asin(sqrt((s-c)/(2*a_new)));
        if less180
            beta = beta_0;
        else
            beta = -beta_0;
        end
        if TOF > TOFmin
            alpha = 2*pi - alpha_0;
        else
            alpha = alpha_0;
        end
        TOF_new = (1/n_new)*((alpha-beta)-(sin(alpha)-sin(beta)));
        diff = abs(TOF_new-TOF);
        %Divergence stopping condition
        if diff > diff2
            fprintf('Calculations started Diverging - Stopping iterations \n')
            a_new = a_bef;
            break
        end
        %Too many Iterations stopping condition
        if iterations > 300
            fprintf('Too many iterations reached, adapt your algorithm for the problem↙
\n')
            break
        end
        diff2 = diff;
        fprintf('diff = %0.12f \n',diff)
        a_initial = a_new + 150;
        iterations = iterations +1;
    end
    aCalc = a_new;
end
```