AE4878 - Mission Geometry and Orbit Design
Part 2 - Coordinate Transformations 2

Ali Nawaz
3ME, LR, Delft University of Technology.

September 2, 2018

1 Introduction

The aim of this assignment is to first verify the parameters tabulated in Table 2-13 of textbook[1]. The results of this is outlined in Section 2 while the script is presented in Section 4.1. Once the script is verified, it is extended to find orbital parameters (a, i) for given orbital requirements. The results of this is outlined in Section 3. Underlying script for the estimations is outlined in Section 4.2

1.1 Co-ordinate System and Parameters Values

Coordinate system used throughout this assignment is illustrated with the aid of Figure 1[2].

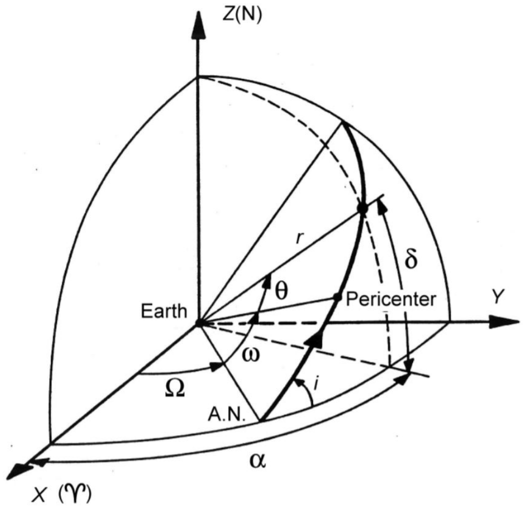

Figure 1: Three dimensional Kepler coordinate frame.

Some of the standard parameters used through out the assignment in outlined in Table 1.

Table 1: Standard parameters used for the assignment.

Parameter	Definition	Value	Unit	Source
μ	Standard gravitational parameter of Earth	3.986004419$\cdot 10^{14}$	m^3/s^2	`en.wikipedia.org/wiki/Standard_gravitational_parameter`
D	Sidereal rotation period of the Earth	86164.10035	s	Orbit & Constellation Design & Management, 2nd printing- Wertz, James R. - pg.81
R_E	Radius of Earth	6378.136$\cdot 10^3$	m	AE4878 Mission Geometry and Orbit Design R. Noomen, TU Delft Lecture - Intro,Slide 16, OCDM parameter list
J2	Dimensionless J2 parameter for Earth	1082.63$\cdot 10^{-6}$	-	Same as above

[1]Orbit Constellation Design Management, 2nd printing- Wertz, James R.
[2]Fundamentals of Astrodynamics - K. Wakker pg.261 `https://www.researchgate.net/publication/272507882_Fundamentals_of_Astrodynamics`

2 Verification of Table 2-13 of OCDM

This section is dedicated to verifying the values presented in Table 2-13 of OCDM[3]. A Matlab script is written to facilitate the verification. Input parameters to the script are the repeat pattern i.e. *j orbits* in *k days*. Furthermore inclination angle is chosen to be 28 deg. Orbits per day follows from the input parameter *j/k*. Table against which the values are verified is presented in Figure 2.

TABLE 2-13. Representative Repeating Ground Track Orbits. A large number of such orbits are possible.

Altitude (km)	Inclination (deg)	Period (min)	Orbits per Day	Repeat Pattern
817.14	28	101.24	14.0	14 orbits/day
701.34	28	98.80	14.33	43 orbits/3 days
645.06	28	97.63	14.50	29 orbits/2 days
562.55	28	95.91	14.75	59 orbits/4 days
546.31	28	95.57	14.8	74 orbits/5 days
482.25	28	94.25	15.0	15 orbits/day

Figure 2: Table 2-13, against which the values from the script are verified.

The procedure used to formulate the script is outlined with the aid of Equation 1[4]. An exception is implemented after the primary initialisation of the semi-major axis a_0. This is explained after the derivations in Equation 1.

Initialise semi- major axis a_0 in meters

$$2\pi\sqrt{\frac{a^3}{\mu}} = \frac{Dk}{j}$$

Above expression gives, initial value for a in meters.

If the altitude for a given initial "a" is negative, then initialise "a" to R_E.

Update the rotation rate of the Earth in deg/sidereal-day

Note that, one sidereal day has D seconds.

$$\dot{L} = 360$$

Update rate of change of Right Ascension of Ascending Node in deg/sidereal-day

$$\dot{\Omega} = \left(-\frac{3}{2}J_2\sqrt{\mu}R_E^2\right)\cdot\left(a^{-3.5}\cos i(1-e^2)^{-2}\frac{180}{\pi}D\right)$$

Update rate of change of argument of pericenter in deg/sidereal-day (1)

$$\dot{\omega} = \left(\frac{3}{4}J_2\sqrt{\mu}R_E^2\right)\cdot\left(a^{-3.5}(5\cos i^2-1)(1-e^2)^{-2}\frac{180}{\pi}D\right)$$

Update rate of change of mean anomaly in deg/sidereal-day

$$\dot{M} = \left(\frac{3}{4}J_2\sqrt{\mu}R_E^2\right)\cdot\left(a^{-3.5}(3\cos i^2-1)(1-e^2)^{-1.5}\frac{180}{\pi}D\right)$$

Mean motion of the satellite in deg/sidereal-day

$$n = \frac{j}{k}\cdot(\dot{L}-\dot{\Omega})-(\dot{\omega}+\dot{M})$$

Update the semi-major axis a in meters.

$$a^3 = \frac{\mu}{\left(\frac{n\pi}{D\cdot 180}\right)^2}$$

Repeat the process starting at $\dot{\Omega}$

[3]Orbit Constellation Design Management, 2nd printing- Wertz, James R. - pg.82
[4]AE4878 Mission Geometry and Orbit Design Lecture - Basics v4-26 Slides - 69 R. Noomen, TU Delft

Equation 1 made this distinction from the lectures slides, including an extra step. The step was to initialise a to R_E, if the initialised semi-major axis results in an altitude below 0m. It was observed during the verification run, that for $-ve$ initialisation of altitude, the program might become unstable. The algorithm is sensitive to $-ve$ altitude initialisation. This occurred for 74 orbits every 5 days scenario, where the value of semi-major axis converged to a 0 value. $-ve$ altitude initialisation was observed for all non-integer repeat patterns(j/k). If initialised a$\geq R_E$, then values of a for both integer and non-integer repeat patterns converge to a positive finite value. Table 2 verifies the altitude output obtained via the script 4.1 against the tabulated values in Table 2. The difference is determined with the aid of Equation 2.

$$\%\text{Difference} = \frac{|\text{Table Values - Estimated Values}|}{\text{Table Values}} \cdot 100\% \tag{2}$$

Table 2: Verification of estimated altitude against altitude in OCDM.

Estimated altitude	**Altitude in Table 2-13**	**Unit**	**% Difference**
817.1666184	817.14	km	0.003257511250
701.2531473	701.34	km	0.01238381749
644.9013908	645.06	km	0.02458828441
562.2876958	562.55	km	0.04662770828
546.0327048	546.31	km	0.05075784019
481.8783074	482.25	km	0.07707466293

A logical stopping criteria would be the difference between the previous and current semi-major axis. However, it was observed that this could result in convergence before reaching the true value. Thereby, the stopping criteria was chosen as the no. of iteration steps. Usually after 5 iteration steps, all the values seemed to converge with the script presented in Section 4.1.

3 Inventory list: semi major axis and inclination angle

This section aims at finding possible options for semi-major axis a and inclination angle i, for given requirements. The requirements imposed are k = 3 (3 day repeat orbits), eccentricity = 0, 200km $\leq$orbital height $\leq$ 1200km and j =(39 - 48) no. of orbits in k days.

Two approaches are implemented. For the first approach J_2 effects are taken into account for perturbations on Ω only. For the second approach J_2 effects are taken into account for perturbations on Ω, ω and M.

Approach 2, is the same as the approach in Section 2. While, approach 1 excludes the factors $\dot{\omega}$ and $\dot{M}$, which in turn has effect on the mean motion, n. Modifications for approach 1 are outlined with the aid of Equation 3.

$$\begin{aligned} \dot{\omega} &= 0 \\ \dot{M} &= 0 \\ n &= \frac{j}{k} \cdot (\dot{L} - \dot{\Omega}) \end{aligned} \tag{3}$$

Matlab script for both approaches is presented in 4.2.

Inputs to the program is listed as follows:

- j = [39, 40, 41, 42, 43, 44, 45, 46, 47, 48] no of orbits in k= 3 days.
- Eccentricity = 0
- Orbital altitude bound [200km,1200km]

Inclination angle i, is varied between 0 and 180 degrees. Using the same process outlined in Section 2, values for semi-major axis is obtained. If the corresponding value of altitude for a given value of semi-major axis is within the given altitude bounds then (a,i) is stored. Otherwise they are discarded. Stopping criteria is again chosen as the number of iterations. In this case 100 iterations are chosen, way more than required. Inclination angle is varied between 0 and 180 degrees, since increasing it further only changes the velocity direction of the satellite while the orbital plane stays the same. A script is presented in Section 4.2.

Results for approach 1 is outlined with the aid of Figure 3, while the results for approach 2 is outlined with the aid of Figure 4. To generate both of these figures, the inclination angle is varied between 0 and 180 degrees with increments of 0.1 degree.

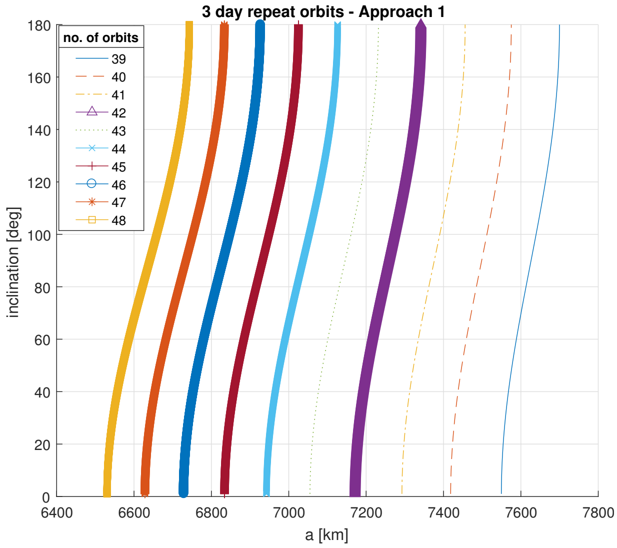

Figure 3: Inclination angle and semi major axis choices with approach 1.

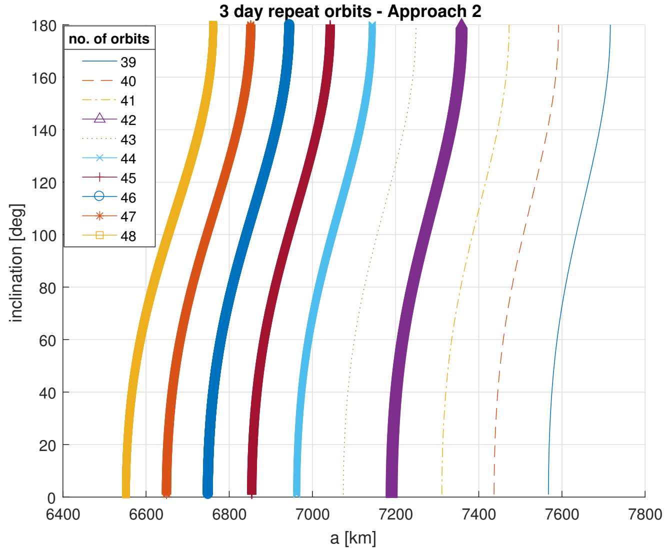

Figure 4: Inclination angle and semi major axis choices with approach 2.

Figures 5 and 6 are presented to visually compare the difference between approach 1 and approach 2. Figure 5 gives an indication of the number of orbits with the aid of legend. However, this gets messy and since from Figures 3 and 4 it is known that increasing the no. of orbits has decreasing effect on the semi major axis values. To provide a clear difference between the two approaches Figure 6 is provided.

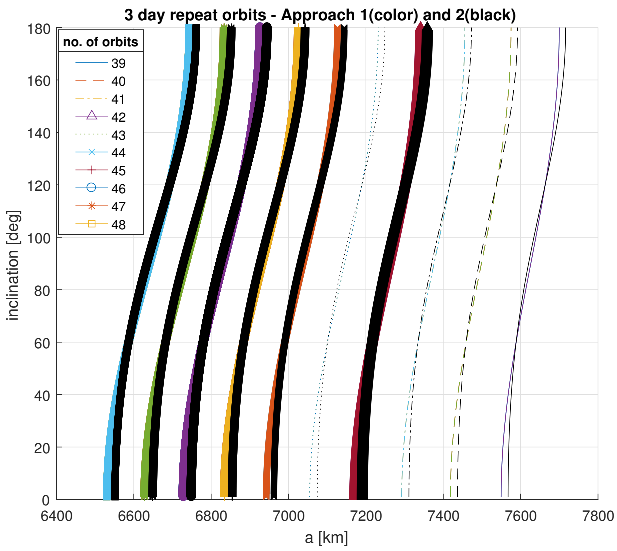

Figure 5: Comparison approach 1 and approach2.

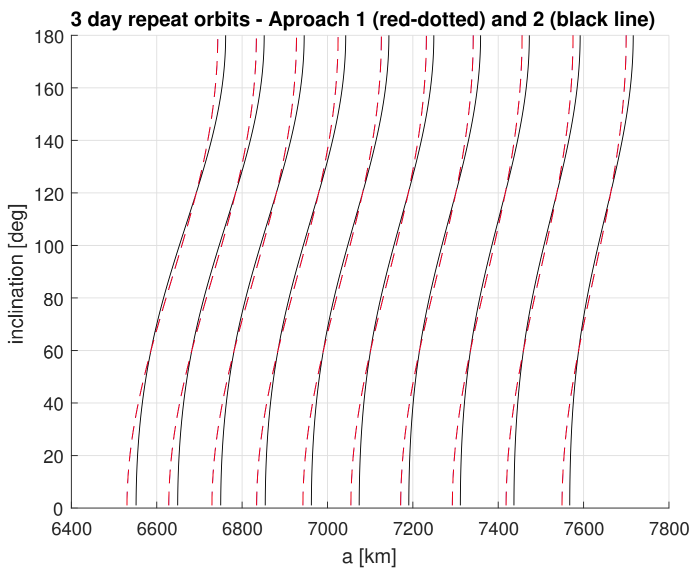

Figure 6: Comparison approach 1 and approach 2. Increasing the no. of orbits, decreases semi-major axis a.

A script is provided in Section 4.2, which is used to facilitate the above estimations. The above results look accurate when verified against Figure 7[5]. From Figure 6, it can be observed that for extremities of inclination angle (orbital plane close to equator), the semi major axis for approach 1 has lower magnitude than semi major axis for approach 2. While close to polar orbit (inclination ≈ 90 deg) the semi major axis has slightly higher magnitude for approach 1 than for approach 2.

[5] AE4878 Mission Geometry and Orbit Design Lecture - Basics v4-26 Slides - 72, R. Noomen, TU Delft

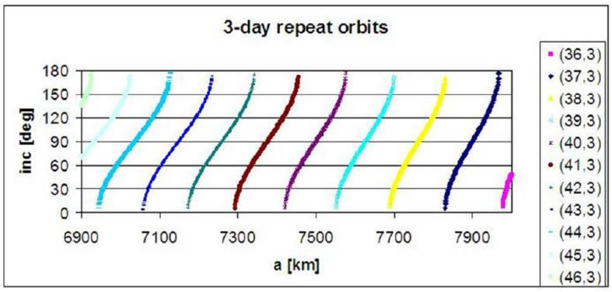

Figure 7: Verification plot of inclination angle vs semi-major axis, for varying j,k.

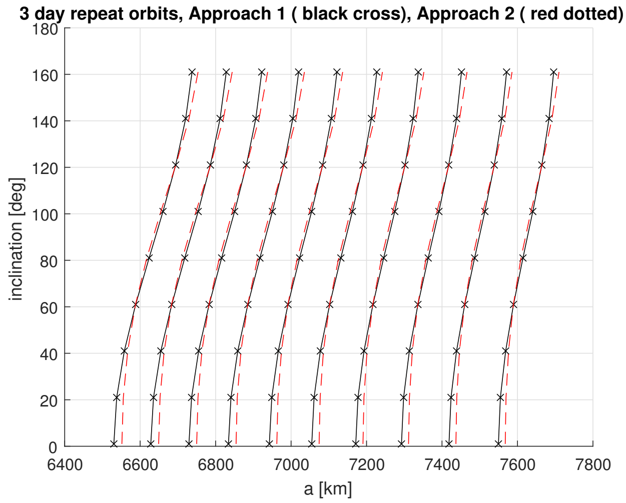

Figure 8: Plot of inclination angle vs semi-major axis, for varying j,k. Inclination angle varies by 20 deg per step.

From Figure 6 it can be further observed that the difference between approach 1 and approach 2 starts to increase for increasing no. of orbits in k days. Table 3 summarises the results obtained via approach 1 and 2. Tabulating all the values of inclination and semi major axis that went into the plots is unrealistic. Thus inclination angle is varied between by 20 deg in the range [0 deg, 180 deg]. The result of this is outlined with the aid of Figure 8. Furthermore, only the extremes of j (i.e. j = 39, 48) are tabulated. Overall it can be observed that increasing the number of j orbits in k days, increases the absolute difference between Approach 1 and 2. Furthermore at extremes of inclination angle the differences are larger. With low differences around 60 and 120 degrees.

Table 3: Values of semi major axis a (in km) for different inclination angles(deg), for j = 39 and 48.

j = 39				j = 48			
Inclination [deg]	**Approach 1 [km]**	**Approach 2 [km]**	**Difference**	**Inclination [deg]**	**Approach 1[km]**	**Approach 2 [km]**	**Difference**
1	7549.521816	7567.554730	18.03291354	1	6530.270403	6551.575369	21.30496515
21	7554.722359	7569.640623	14.91826350	21	6537.839809	6555.432868	17.59305903
41	7568.678827	7576.301729	7.622901676	41	6558.072138	6567.017463	8.945324879
61	7589.5186594	7589.1654213	0.3532381146	61	6588.069289	6587.657870	0.4114191989
81	7614.5287344	7609.2744195	5.254314940	81	6623.746255	6617.682252	6.064003099
101	7640.5789220	7635.6791494	4.899772706	101	6660.552147	6654.9515072	5.600639510
121	7664.5478931	7664.8926463	0.3447531020	121	6694.1148732	6694.5054559	0.3905826730
141	7683.6826396	7691.5713970	7.888757442	141	6720.7103144	6729.5848609	8.874546532
161	7695.8634276	7710.0901103	14.22668270	161	6737.5522908	6753.4854020	15.93311125

4 Matlab Scipt

4.1 Verification of Table 2-13

```
%% AE4878 - Mission Geometry and Orbit Design
%% HW Assignment 2 - BASICS 7
%% Student info - Ali Nawaz, 4276477
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Question 1 : Verification of Table 2-13 from OCDM
% Standard parameters

mu = 3.986004419*10^(14); % [m^3 s^-2] standard gravitational parameter of Earth from Wikipedia
D = 86164.10035; % sidereal rotation period of the Earth [s]. Source OCDM pg.81
RE = 6378.136*10^(3); % radius of Earth [m]. Source: OCDM parameter list, intro lecture, ...
    slide 16..
J2 = 1082.63*10^(-6); % Dimensionless J2 parameter. Source: Lecture intro, slide 16

% Verification parameters
```

```
iv_deg = 28; % inclination angle [deg] for verification
iv_rad = deg2rad(iv_deg); % inclination angle [rad] for verification

jv = [ 14, 43, 29, 59, 74, 15]; % integer number of orbits for integer numner of days, k.
kv = [ 1, 3, 2, 4, 5, 1]; % integar number of days for integar number of orbits, j.

ev = 0; % eccentricity = 0 for verification
repeat_pattern = jv./kv; % repeat pattern, j orbits in k days

% Algorithm for verification of Table 2-13 of OCDM
a = [];
a0_list = [];
for z = 1:length(jv)
    % initialize semi major axis a0
    a0 = (mu^(1/3) )* ( 2*pi*jv(z)/D*kv(z))^(-2/3); % [m]
    % List of a0
    a0_list = [a0_list,a0];
    % Avoiding -ve altitude initialisation for non-integar repeat patterns
    if (a0-RE) < 0
        a0 = RE;
    end

    acc = 10^-3; % accuracy [m];
    err = 10000; % Initializes while loop.
    iter = 0;
    a(iter+1,z) = [a0];
    while err > acc;
        iter = iter +1;
        L_dot_deg = 360; % Rotation rate of the Earth in deg/day
        L_dot_rad = deg2rad(L_dot_deg); % Rotation rate of the Earth in rad/day

        Omega_dot = ...
            -(3/2)*J2*sqrt(mu)*(RE^2)*(a(iter,z)^-3.5)*cos(iv_rad)*((1-ev^2)^(-2))*(180/pi)*D; ...
            %  Rate of change of Right ascension of ascending node in deg/sidereal day
        omega_dot = ...
            (3/4)*J2*sqrt(mu)*(RE^2)*(a(iter,z)^-3.5)*(5*(cos(iv_rad)^2)-1)*((1-ev^2)^(-2))*(180/pi)*D; ...
            % Rate of change of argument of pericenter in deg/siderealday
        M_dot = (3/4)*J2*sqrt(mu)*(RE^2)*(a(iter,z)^-3.5)*(3*(cos(iv_rad)^2) ...
            -1)*((1-ev^2)^(-1.5))*(180/pi)*D; % Rate of change of Mean anomaly [deg/siderealday]

        n = (jv(z)/kv(z)) * (L_dot_deg - Omega_dot) - (omega_dot + M_dot); % Mean motion of ...
            the satellite in orbit [deg/sidreal day]

        a(iter+1,z) = ( mu/ ( ( (n/D)*(pi/180))^2 ))^(1/3); % Updated semi-major axis [m];

        if iter> 100000
            % Infinite loop avoidance criteria
            break;
        end
%           err = a(iter+1,z) - a(iter,z);
    end
end

% Orbit altitude in km
altitude = (a(end,:) - RE)*10^(-3); % [km]
% Values of altitude from Table 2-13 of the textbook
verification_alt = [817.14, 701.34, 645.06, 562.55, 546.31, 482.25]; % [km]

% Difference of the estimated altitude and the tabulated altitude.
alt_diff = ( abs(altitude- verification_alt)./verification_alt)*100; % Accuracy of estimation ...
    in %
```

4.2 Semi major axis and inclination angle inventory script

```
%% Question 2

% Inputs to the program
j = [ 39, 40, 41, 42, 43, 44, 45, 46, 47, 48]; % integer number of orbits for integar numner ...
    of days, k.
k = 3;  % integar number of days for integar number of orbits, j.

```

```
e = 0; % eccentricity = 0

lb = 200*10^(3); % Lower bound on orbital altitude [m]
ub = 1200*10^(3); % Upper bound on orbital altitude [m]

% Outputs are semi-major axis a and inclination angle i.

% Algorithm for approach 1 and approach 2

result = []; % initialising result matrix
for z = 1:length(j)
    disp(['Running j: ',num2str(z),', out of: ',num2str(length(j))]) % Display the progress

    for inc = 1:0.1:180
        a = []; % initialise the list of semi-major axis a
        a0_list = []; % list of initial semi-major axis a0
        inc_rad = deg2rad(inc); % Converting inclination from degrees to radians
        % initialize semi major axis a0
        a0 = (mu^(1/3) )* ( 2*pi*j(z)/D*k)^(-2/3); % [m]
        % List of a0
        a0_list = [a0_list,a0];
        % Avoiding -ve altitude initialisation for non-integar repeat patterns
        if (a0-RE) < 0
            a0 = RE;
        end

        acc = 10^-3; % accuracy [m];
        err = 10000; % Initializes while loop.
        iter = 0;   % iteration number
        a(iter+1,z) = [a0];
        while err > acc;
            iter = iter +1;
            L_dot_deg = 360; % Rotation rate of the Earth in deg/day
            L_dot_rad = deg2rad(L_dot_deg); % Rotation rate of the Earth in rad/day
            % Approach 1
            Omega_dot = ...
                -(3/2)*J2*sqrt(mu)*(RE^2)*(a(iter,z)^-3.5)*cos(inc_rad)*((1-e^2)^(-2))*(180/pi)*D; ...
                %  Rate of change of Right ascension of ascending node in deg/sidereal day
            omega_dot = 0; % Rate of change of argument of pericenter in deg/siderealday
            M_dot = 0; % Rate of change of Mean anomaly [deg/siderealday]


            n = (j(z)/k) * (L_dot_deg - Omega_dot) - (omega_dot + M_dot); % Mean motion of ...
                the satellite in orbit [deg/sidreal day]

            a(iter+1,z) = ( mu/ ( ( (n/D)*(pi/180))^2 ))^(1/3); % Updated semi-major axis [m];

            % Approach 2
            Omega_dot = ...
                -(3/2)*J2*sqrt(mu)*(RE^2)*(a(iter,z)^-3.5)*cos(inc_rad)*((1-e^2)^(-2))*(180/pi)*D; ...
                %  Rate of change of Right ascension of ascending node in deg/sidereal day
            omega_dot = ...
                (3/4)*J2*sqrt(mu)*(RE^2)*(a(iter,z)^-3.5)*(5*(cos(inc_rad)^2)-1)*((1-e^2)^(-2))*(180/pi)*D; .
                % Rate of change of argument of pericenter in deg/siderealday
            M_dot = (3/4)*J2*sqrt(mu)*(RE^2)*(a(iter,z)^-3.5)*(3*(cos(inc_rad)^2) ...
                -1)*((1-e^2)^(-1.5))*(180/pi)*D; % Rate of change of Mean anomaly ...
                [deg/siderealday]


            n = (j(z)/k) * (L_dot_deg - Omega_dot) - (omega_dot + M_dot); % Mean motion of ...
                the satellite in orbit [deg/sidreal day]

            a(iter+1,z) = ( mu/ ( ( (n/D)*(pi/180))^2 ))^(1/3); % Updated semi-major axis [m];

            if iter> 100
                % Infinite loop avoidance criteria
                break;
            end
    %         err = a(iter+1,z) - a(iter,z);
        end

        % Store results if the altitude is within desirable bounds
        altitude = a(end) - RE;
        if altitude ≥ lb | altitude ≤ ub
            result = [result;[j(z), inc, a(end)*10^-3, altitude*10^(-3)]]; % Store  j [no. of ...
                orbits], inclination angle [deg], a[km] and altitude [km] if altitude limit ...
```

```
                 is satisfied
        end
    end
end
%% Plotting results

% Save results for each j on a seperate page to facilitate plotting.
pp = [];
for p = 1:length(j)
    entry = 0;
    for r = 1:length(result(:,1))
        if result(r,1) == j(p)
            entry = entry +1;
            pp(entry,:,p) = result(r,:);
        end
    end
end

% Plot figure
figure(1)
hold on
plot(pp(:,3,1),pp(:,2,1),'-',pp(:,3,2),pp(:,2,2),'--',pp(:,3,3),pp(:,2,3),...
'-.',pp(:,3,4),pp(:,2,4),'-^',pp(:,3,5),pp(:,2,5),':',pp(:,3,6),pp(:,2,6),...
'-x',pp(:,3,7),pp(:,2,7),'-+',pp(:,3,8),pp(:,2,8),'-o',pp(:,3,9),pp(:,2,9),...
'-*',pp(:,3,10),pp(:,2,10),'-s');
hleg = legend('39','40','41','42','43','44','45','46','47','48','Location','Best');
htitle = get(hleg,'Title');
set(htitle,'String','no. of orbits');
xlabel('a [km]');
ylabel('inclination [deg]');
title ('3 day repeat orbits')
grid on
hold off
```