AA528: Homework 4

Berit Syltebo

February 2023

Problem 12.6

Assume an Earth-relative orbit is given by the initial orbit elements $a = 7300$ km, $e = 0.05$, $i = 42°$, $\Omega = 0°$, $\omega = 45°$, and $M_0 = 0°$. Assume the disturbance acceleration is solely due to the J_2 gravitational acceleration given in Eq. (11.57).

(a) Set up a numerical simulation to solve the true nonlinear motion $\{x(t), \dot{x}(t)\}$ using Eq. (12.1) for 10 orbits.

(b) Translate the $\{x(t), \dot{x}(t)\}$ coordinates into the corresponding classical orbit elements $\{a, e, i, \Omega, \omega, M_0\}$.

(c) Compare the numerically computed $\boldsymbol{e}(t)$ motion to the analytically predicted instantaneous orbit element variations in Eq. (12.87) and the average orbit element variations in Eq. (12.88).

Part (a) Analysis and Solution

Eq. (12.1) is provided in Equation 1 of this report. It is specified in the problem statement that the disturbance acceleration is solely due to the J_2 gravitational acceleration such that $\boldsymbol{a}_d = \boldsymbol{a}_{J_2}$, thus explaining the additional condition shown in Equation 1.

$$\ddot{\boldsymbol{r}} = -\frac{\mu}{r^3}\boldsymbol{r} + \boldsymbol{a}_d \qquad \boldsymbol{a}_d = \boldsymbol{a}_{J_2} \tag{1}$$

The first step in solving Ordinary Differential Equation (ODE) above was to find the initial conditions once setting a coordinate frame. For simplicity, the coordinate frame defining the the ODE was set to the Earth-Centered Inertial (ECI) frame. From this frame, the state was defined as:

$$x(t) = \begin{bmatrix} r_{x_{ECI}} & r_{y_{ECI}} & r_{z_{ECI}} & v_{x_{ECI}} & v_{y_{ECI}} & v_{z_{ECI}} \end{bmatrix}^T$$

Short-hand notation in this report will exclude the "ECI" subscript. The initial conditions vector was then defined:

$$x(0) = \begin{bmatrix} r_{x0} & r_{y0} & r_{z0} & v_{x0} & v_{y0} & v_{z0} \end{bmatrix}^T$$

In order to find these initial quantities, the following equations were used to solve for the position and velocity vectors. Both Equatiosn 2 and 3 are in the orbit frame $\mathcal{O} : \{\hat{i}_r, \hat{i}_\theta, \hat{i}_h\}$ but are simple to work with in this problem and only require a Direction Cosine Matrix (DCM) to convert to the proper frame.

$$\boldsymbol{r} = r\hat{i}_r \tag{2}$$

$$\boldsymbol{v} = \dot{r}\hat{i}_r + r\dot{f}\hat{i}_\theta \tag{3}$$

The variables r_0, $\dot{r}_0$, and $\dot{f}_0$ were found via the following computations:

$$r_0 = \frac{p}{1+e} \qquad p = a(1-e^2)$$

$$\dot{f}_0 = \frac{\sqrt{\mu p}}{r^2}$$

$$\dot{r}_0 = 0$$

The resulting initial conditions in the $\mathcal{O}$ frame are:

$$\boldsymbol{r}_0 = \frac{p}{1+e}\hat{i}_r + 0\hat{i}_\theta + 0\hat{i}_h$$

$$\boldsymbol{v}_0 = 0\hat{i}_r \frac{p}{1+e}\frac{\sqrt{\mu p}}{r^2}\hat{i}_\theta + 0\hat{i}_h$$

In order to transform these vectors into the ECI frame, they are multiplied by the DCM, defined as $[NO]$, below. In the book, this is the transpose of Eq. (9.50) with Ω already substituted to simplify the matrix. For this case θ is equivalent to the initial ω value, 45°.

$$[NO] = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta\cos i & \cos\theta\cos i & -\sin i \\ \sin\theta\sin i & \cos\theta\sin i & \cos i \end{bmatrix}$$

Numerically implementing this DCM on the initial position and velocity vectors formerly defined in the $\mathcal{O}$ frame, the initial conditions defining the vector, $x(0)$ are found:

$$x(0) = \begin{bmatrix} 4903.8 & 3644.2 & 3281.3 & -5.5 & 4.1 & 3.7 \end{bmatrix}^T$$

The next step was to come up with the ODE in terms of the defined state such that...

$$\dot{x}(t) = Ax(t) \tag{4}$$

$$A \in \mathbb{R}^{n\times m}, \quad x(t) \in \mathbb{R}^m, \quad \dot{x}(t) \in \mathbb{R}^n$$

Expanding Equation 4 and combining A and $x(t)$...

$$\dot{x}(t) = \begin{bmatrix} \dot{r}_x \\ \dot{r}_y \\ \dot{r}_z \\ \dot{v}_x \\ \dot{v}_y \\ \dot{v}_z \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ v_z \\ \ddot{r}_x \\ \ddot{r}_y \\ \ddot{r}_z \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ v_z \\ -\frac{\mu}{r^3}r_x + a_{J_{2x}} \\ -\frac{\mu}{r^3}r_y + a_{J_{2y}} \\ -\frac{\mu}{r^3}r_z + a_{J_{2z}} \end{bmatrix}$$

The last notable step was to define $\boldsymbol{a}_{J_2}$ so that the expression above could be complete. Equation 5 below completes the last three indices of the ODE and results in an expression purely in terms of the variables of $x(t)$. With this last computation, the ODE is ready to be solved. Note: notation differs slightly so that $x = r_x$, $y = r_y$, and $z = r_z$.

$$\boldsymbol{a}_{J_2} = -\frac{3}{2} J_2 \left(\frac{\mu}{r^2}\right) \left(\frac{r_{eq}}{r}\right)^2 \begin{pmatrix} \left(1 - 5\left(\frac{z}{r}\right)^2\right)\frac{x}{r} \\ \left(1 - 5\left(\frac{z}{r}\right)^2\right)\frac{y}{r} \\ \left(3 - 5\left(\frac{z}{r}\right)^2\right)\frac{z}{r} \end{pmatrix} \tag{5}$$

Using MATLAB's ODE solver, `ode89()`, a simulation is obtained for 10 orbits. The time domain was determined by calculating the period of the orbit and multiplying by 10: 62,073 seconds. The computed solution is plotted in Figure 1.

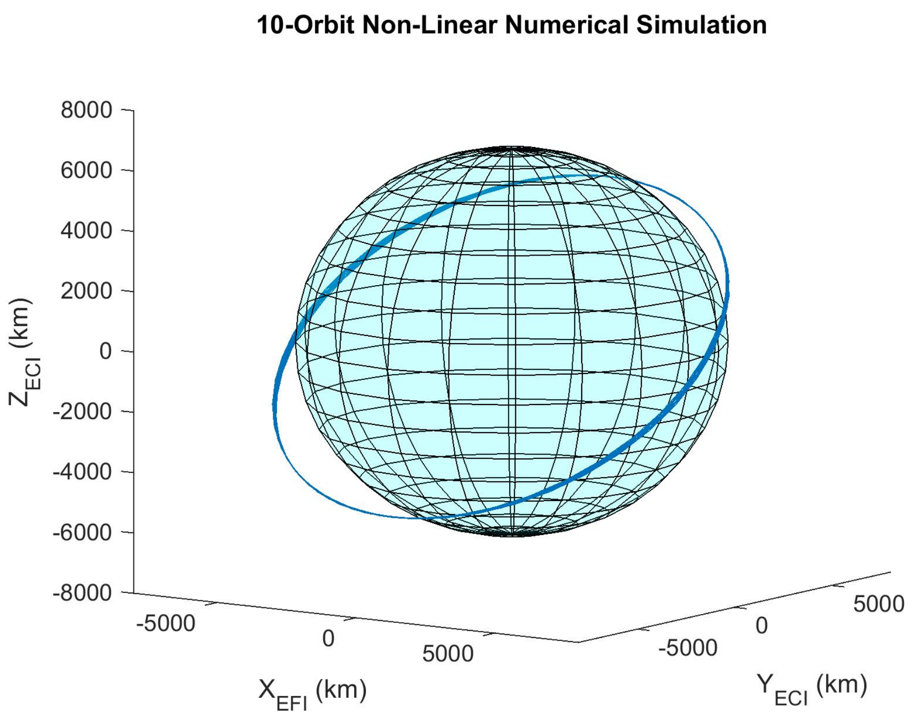

Figure 1: Non-linear simulation of satellite orbiting Earth 10 times including perturbation considerations with J_2

Part (b) Analysis

In order to make the conversion between position and velocity, and the classical orbital elements, the individual components of position in velocity in the ECI frame at each time stamp are considered. These elements can be defined by the following vector:

$$x(t) = \begin{bmatrix} r_x(t) & r_y(t) & r_z(t) & v_x(t) & v_y(t) & v_z(t) \end{bmatrix}^T$$

The first orbital element computed was the semi-major axis, a. This was done by using the Conservation of Energy principle and the vis-viva equation defined in Equation 6. It is assumed that r and v are the norm of their respective vectors. Rearranging and solving for a yields that semi-major axis at that given time.

$$\frac{v(t)^2}{2} - \frac{\mu}{r(t)} = -\frac{\mu}{2a(t)} \tag{6}$$

Next, the eccentricity was computed using Equation 7. The angular momentum, also as a function of time, is defined shortly after to complete the analysis for eccentricity. The norm is taken as the final parameter.

$$\boldsymbol{e}(t) = \frac{\boldsymbol{v}(t) \times \boldsymbol{h}(t)}{\mu} - \frac{\boldsymbol{r}(t)}{r(t)} \tag{7}$$

$$\boldsymbol{h}(t) = \boldsymbol{r}(t) \times \boldsymbol{v}(t)$$

The three parameters Ω, ω, and i are determined by finding the DCM to get from the ECI frame to the perifocal coordinate frame, $\mathcal{P} : \{\hat{i}_e, \hat{i}_p, \hat{i}_h\}$. The elements of this matrix, $[C]$ are used to compute these parameters. This process is described below and the final results are given in Equations 8-10.

$$\hat{i}_e(t) = \boldsymbol{e}(t)/e(t) = C_{11}\hat{i}_x + C_{12}\hat{i}_y + C_{13}\hat{i}_z$$
$$\hat{i}_p(t) = \hat{i}_h(t) \times \hat{i}_e(t) = C_{21}\hat{i}_x + C_{22}\hat{i}_y + C_{23}\hat{i}_z$$
$$\hat{i}_h(t) = \boldsymbol{h}(t)/h(t) = C_{31}\hat{i}_x + C_{32}\hat{i}_y + C_{33}\hat{i}_z$$

$$\Omega(t) = \tan^{-1}\left(\frac{C_{31}}{-C_{32}}\right) \tag{8}$$

$$i(t) = \cos^{-1}(C_{33}) \tag{9}$$

$$\omega(t) = \tan^{-1}\left(\frac{C_{13}}{C_{23}}\right) \tag{10}$$

Lastly, the initial mean anomaly is found using Equation 11. The initial eccentricity is computed by taking the cross product of the initial velocity and angular momentum.

$$M_0(t) = \cos^{-1}\left(\frac{\boldsymbol{e}(t) \cdot \boldsymbol{e}_0}{|\boldsymbol{e}(t)||\boldsymbol{e}_0|}\right) \tag{11}$$

Iterating over all times for Equations 5-11 results in a complete data set for the satellite's motion over the course of ten orbits in terms or the six classical orbit elements.

Part (c) Analysis

The analytical predicted instantaneous orbit element variations are defined in Equations 12-17. Derivations of some parameters like the semi-minor axis, b, and η are not defined in this report as they are very simple computations. These are provided in the attached code.

Using MATLAB's ODE solver again, `ode89()`, these ODE's are integrated across the same time domain used in part (a) of this problem. The initial conditions are merely the parameters given in the problem statement.

$$\frac{d\Omega}{dt} = -3J_2 n \frac{a^2}{br}\left(\frac{r_{eq}}{r}\right)^2 \sin^2\theta \cos i \tag{12}$$

$$\frac{di}{dt} = -\frac{3}{4}J_2 n \frac{a^2}{br}\left(\frac{r_{eq}}{r}\right)^2 \sin(2\theta)\sin(2i) \tag{13}$$

$$\begin{aligned}\frac{d\omega}{dt} = \frac{3}{2}J_2 n \frac{p}{r^2 e\eta^3}\left(\frac{r_{eq}}{r}\right)^2 [&2re\cos^2 i \sin^2\theta \\ &- (p+r)\sin f \sin^2 i \sin(2\theta) \\ &+ p\cos f(1 - 3\sin^2 i \sin^2\theta)]\end{aligned} \tag{14}$$

$$\frac{da}{dt} = -3J_2 n \frac{a^4}{br^2}\left(\frac{r_{eq}}{r}\right)^2 \left[e\sin f(1 - 3\sin^2\theta\sin^2 i) + \frac{p}{r}\sin(2\theta)\sin^2 i\right] \tag{15}$$

$$\frac{de}{dt} = -\frac{3}{2}J_2 n\frac{a^2}{br}\left(\frac{r_{eq}}{r}\right)^2 [\frac{p}{r}\sin f(1-3\sin^2\theta\sin^2 i)$$
$$+ (e + cos f(2 + e\cos f))\sin(2\theta)\sin^2 i] \quad (16)$$

$$\frac{M_0}{dt} = \frac{3}{2}J_2 n\frac{p}{r^2 e\eta^2}\left(\frac{r_{eq}}{r}\right)^2 [(p+r)\sin f\sin^2 i\sin(2\theta)$$
$$+ (2re - p\cos f)(1 - 3\sin^2 i\sin^2\theta)] \quad (17)$$

The average orbit element variations were then found by integrating over Equations 18-23 just like with the instantaneous variations. The same initial conditions and time frame were used.

$$\frac{d\bar{\Omega}}{dt} = -\frac{3}{2}J_2 n\left(\frac{r_{eq}}{p}\right)^2 \cos i \quad (18)$$

$$\frac{d\bar{i}}{dt} = 0 \quad (19)$$

$$\frac{d\bar{\omega}}{dt} = \frac{3}{4}J_2 n\left(\frac{r_{eq}}{p}\right)^2 (5\cos^2 i - 1) \quad (20)$$

$$\frac{d\bar{a}}{dt} = 0 \quad (21)$$

$$\frac{d\bar{e}}{dt} = 0 \quad (22)$$

$$\frac{d\bar{M}_0}{dt} = \frac{3}{4}J_2 n\left(\frac{r_{eq}}{p}\right)^2 \sqrt{1-e^2}(3\cos^2 i - 1) \quad (23)$$

Parts (b) and (c) Solutions

The results of the analysis conducted in parts (b) and (c) can be found in Figures 2 and 3. Each orbital element is plotted separately and compares the numerical derivation from part (b) with the instantaneous analytical and mean variations.

Some parameters are nearly identical between numerical and analytical approaches (like longitude of ascending node and eccentricity) while the analytical solutions for some tend to diverge faster from the initial parameters (namely inclination and semi-major axis). Despite some discrepancies, they all tend to maintain the same shape between methods although some seem shifted like argument of periapsis and mean anomaly. The mean variations tend to do a better job at capturing the long-term effects of perturbation than its instantaneous counterpart and follows the mean of the numerical solution much better.

The code used to produce these results is provided at the end of this report.

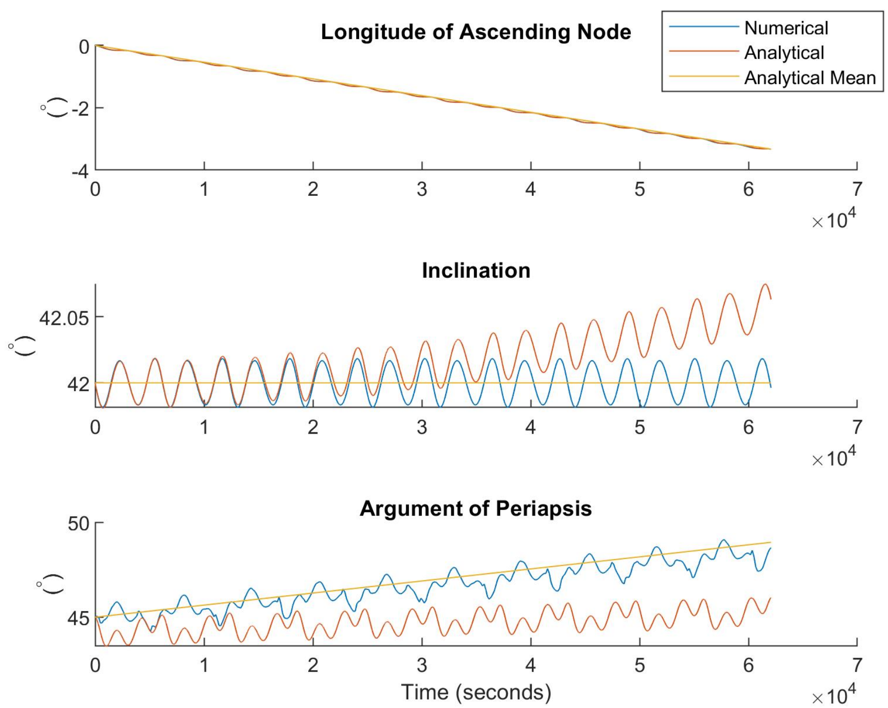

Figure 2: Comparison of numerical and analytical analyses for the longitude of ascending node, inclination, and argument of periapsis over all 10 orbits

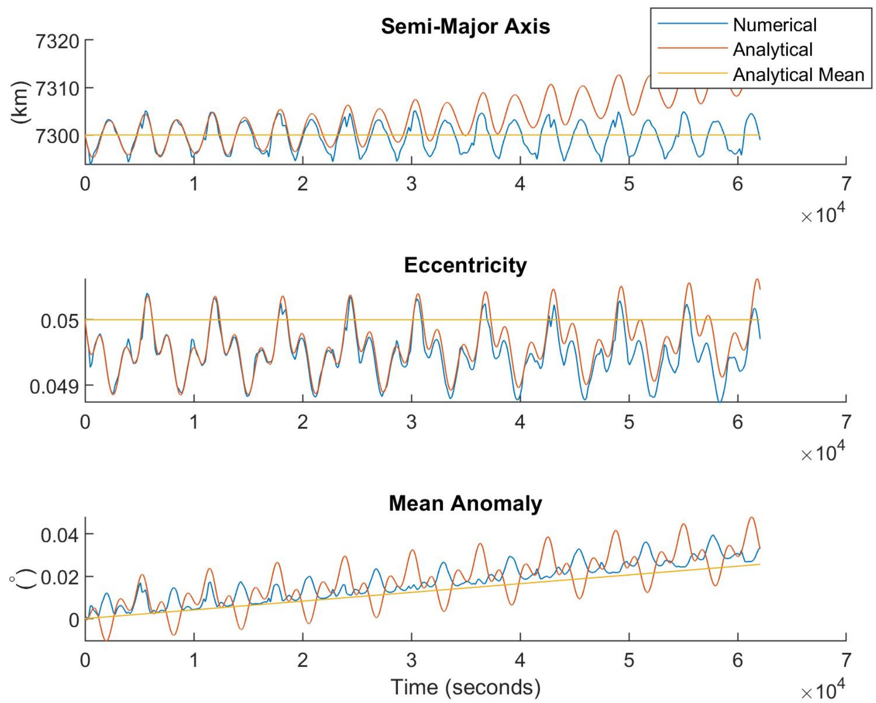

Figure 3: Comparison of numerical and analytical analyses for the semi-major axis, eccentricity, and mean anomaly over all 10 orbits

```
clear all; close all; clc;

%% Useful Equations

% r_ddot = -mu/r^3 * r + a_d;
% rdot_hat = rdot ir + r fdot it
% rdot = r^2 fdot e sin(f) / p


%% Initial Orbital Elements

a = 7300; % km
e = 0.05;
i = 42; % degrees
Omega = 0; % degrees
omega = 45; % degrees
M0 = 0; % degrees


%% Initial Conditions

G = 6.674 * 10^(-20); % km^3/kg-s^2
m1 = 5.97219 * 10^(24); % kg
mu = G * m1;

% Conversion matrix to Cartesian
theta0 = omega;
ON = [cosd(theta0) sind(theta0)*cosd(i) sind(theta0)*sind(i);...
    -sind(theta0) cosd(theta0)*cosd(i) cosd(theta0)*sind(i);...
    0 -sind(i) cosd(i)];
NO = ON';

% Position IC's
p = a*(1-e^2);
r_O0 = [p/(1+e);0;0];
r_N0 = NO * r_O0;

% Velocity IC's
f0 = 0;
r = r_O0(1);
f_dot = sqrt(mu * p) / r^2;
v_O0 = [0;r*f_dot;0];
v_N0 = NO * v_O0;


%% Implementation

tfinal = 10 * (2*pi*sqrt(a^3/mu));
tspan = [0 tfinal];
x0 = [r_N0;v_N0];
```

```
[t,x] = ode89(@vdp1,tspan,x0);

r_earth = 6378; % km
[X,Y,Z] = sphere;
X2 = X * r_earth;
Y2 = Y * r_earth;
Z2 = Z * r_earth;

figure(1)
hold on
plot3(x(:,1),x(:,2),x(:,3))
surf(X2,Y2,Z2,'FaceAlpha',0.1,'FaceColor','c')
xlim([-8000 8000])
ylim([-8000 8000])
zlim([-8000 8000])
view(35,10)
hold off
title('10-Orbit Non-Linear Numerical Simulation')
xlabel('X_{EFI} (km)')
ylabel('Y_{ECI} (km)')
zlabel('Z_{ECI} (km)')


%% Data Conversion to Classical Orbital Elements

h = cross(r_N0,v_N0);
e_i = cross(v_N0,h);

a = [];
e = [];
Omega = [];
inc = [];
omega = [];
M0 = [];
for i = 1:length(x)

    % Semi-major axis, a
    r_i = sqrt(x(i,1)^2 + x(i,2)^2 + x(i,3)^2);
    v_i = sqrt(x(i,4)^2 + x(i,5)^2 + x(i,6)^2);
    a(i) = (2/r_i - v_i^2/mu)^(-1);

    % Eccentricity, e
    r_t = [x(i,1);x(i,2);x(i,3)];
    v_t = [x(i,4);x(i,5);x(i,6)];
    h_vec = cross(r_t,v_t);
    e_vec = cross(v_t,h_vec)/mu - r_t/r_i;
    e(i) = norm(e_vec);

    % Angles, Omega omega and i, from DCM
```

```
    i_e = e_vec/e(i);
    i_h = h_vec/norm(h_vec);
    i_p = cross(i_h,i_e);
    DCM = [i_e(1) i_e(2) i_e(3);...
        i_p(1) i_p(2) i_p(3);...
        i_h(1) i_h(2) i_h(3)];
    Omega(i) = atand(-DCM(3,1)/DCM(3,2));
    inc(i) = acosd(DCM(3,3));
    omega(i) = atand(DCM(1,3)/DCM(2,3));

    % Mean anomaly
    M0(i) = acos(dot(e_vec,e_i)/(norm(e_vec)*norm(e_i)));

end


%% Comparison of numerical and analytical techniques for e(t)

a_init = 7300; % km
e_init = 0.05;
i_init = 42; % degrees
Omega_init = 0; % degrees
omega_init = 45; % degrees
M0_init = 0; % degrees

e0 = [Omega_init;i_init;omega_init;a_init;e_init;M0_init];
[t_a,e_bold] = ode89(@analytical,tspan,e0);
[t_bar,e_bar] = ode89(@ebar,tspan,e0);

% Plot Omega, i, and omega
figure(2)
tiledlayout(3,1)
nexttile
hold on
plot(t,Omega)
plot(t_a,real(e_bold(:,1)))
plot(t_bar,e_bar(:,1))
title('Longitude of Ascending Node')
ylabel('(^\circ)')
legend('Numerical','Analytical','Analytical Mean')
nexttile
hold on
plot(t,inc)
plot(t_a,real(e_bold(:,2)))
plot(t_bar,e_bar(:,2))
title('Inclination')
ylabel('(^\circ)')
nexttile
hold on
plot(t,omega)
```

```
plot(t_a,real(e_bold(:,3)))
plot(t_bar,e_bar(:,3))
title('Argument of Periapsis')
ylabel('(^\circ)')
xlabel('Time (seconds)')

% Plot a, e, and M0
figure(3)
tiledlayout(3,1)
nexttile
hold on
plot(t,a)
plot(t_a,real(e_bold(:,4)))
plot(t_bar,e_bar(:,4))
title('Semi-Major Axis')
ylabel('(km)')
legend('Numerical','Analytical','Analytical Mean')
nexttile
hold on
plot(t,e)
plot(t_a,real(e_bold(:,5)))
plot(t_bar,e_bar(:,5))
title('Eccentricity')
nexttile
hold on
plot(t,M0)
plot(t_a,real(e_bold(:,6)))
plot(t_bar,e_bar(:,6))
title('Mean Anomaly')
ylabel('(^\circ)')
xlabel('Time (seconds)')


%% Functions

function dxdt = vdp1(t,x)

    J2 = 1082.63*10^(-6);

    G = 6.674 * 10^(-20); % km^3/kg-s^2
    m1 = 5.97219 * 10^(24); % kg
    mu = G * m1;

    r_x = x(1); r_y = x(2); r_z = x(3);
    r = sqrt(x(1)^2 + x(2)^2 + x(3)^2);
    r_eq = 6371; % km

    a_J2_coeff = -3/2*J2*(mu/r^2)*(r_eq/r)^2;
    a_J2_comps = [(1-5*(x(3)/r)^2)*(x(1)/r);...
        (1-5*(x(3)/r)^2)*(x(2)/r);(3-5*(x(3)/r)^2)*(x(3)/r)];
```

```
    a_J2 = a_J2_coeff * a_J2_comps;

    v1 = x(4);
    v2 = x(5);
    v3 = x(6);

    dxdt = [v1;v2;v3;-mu/r^3*r_x+a_J2(1);...
        -mu/r^3*r_y+a_J2(2);...
        -mu/r^3*r_z+a_J2(3)];

end

function deboldt = analytical(t,x)

    G = 6.674 * 10^(-20); % km^3/kg-s^2
    m1 = 5.97219 * 10^(24); % kg
    mu = G * m1;

    Omega = x(1);
    i = x(2);
    omega = x(3);
    a = x(4);
    e = x(5);
    M0 = x(6);

    J2 = 1082.63*10^(-6);
    r_eq = 6371; % km

    n = sqrt(mu/a^3);
    b = a*sqrt(1-e^2);

    p = a*(1-e^2);
    eta = sqrt(1-e^2);

    term1 = M0 + n*t;
    E_pre = 0:0.01:100;
    tolerance = 10^(-2);
    for j = 1:length(E_pre)
        term2 = E_pre(j) - e*sin(E_pre(j));
        if abs(term1-term2) <= tolerance
            E = E_pre(j);
            break
        end
    end

    f = rad2deg(2*atan(sqrt((1+e)/(1-e))*tan(E/2)));
    theta = omega + f;
    r = a * (1 - e*cos(E));

    dOdt = -3*J2*n* a^2/(b*r) * (r_eq/r)^2 *...
```

```
        sind(theta)^2 * cosd(i) * 180/pi;
    didt = -3/4*J2*n* a^2/(b*r) * (r_eq/r)^2 *...
        sind(2*theta) * sind(2*i) * 180/pi;
    dodt = 3/2*J2*n* p/(r^2*e*eta^3) * (r_eq/r)^2 *...
        (2*r*e*cosd(i)^2*sind(theta)^2 - ...
        (p+r)*sind(f)*sind(i)^2*sind(2*theta) + ...
        p*cosd(f)*(1-3*sin(i)^2*sind(theta)^2)) * 180/pi;
    dadt = -3*J2*n * a^4/(b*r^2) * (r_eq/r)^2 *...
        (e*sind(f)*(1-3*sind(theta)^2*sind(i)^2) +...
        p/r*sind(2*theta)*sind(i)^2);
    dedt = -3/2*J2*n * a^2/(b*r) * (r_eq/r)^2 *...
        (p/r*sind(f)*(1-3*sind(theta)^2*sind(i)^2) +...
        (e+cosd(f)*(2+e*cosd(f)))*sind(2*theta)*sind(i)^2);
    dM0dt = 3/2*J2*n * p/(r^2*e*eta^2) * (r_eq/r)^2 *...
        ((p+r)*sind(f)*sind(i)^2*sind(2*theta) +...
        (2*r*e-p*cosd(f))*(1-3*sind(i)^2*sind(theta)^2));

    deboldt = [dOdt;didt;dodt;dadt;dedt;dM0dt];

end

function deboldt = ebar(t,x)

    G = 6.674 * 10^(-20); % km^3/kg-s^2
    m1 = 5.97219 * 10^(24); % kg
    mu = G * m1;

    Omega = x(1);
    i = x(2);
    omega = x(3);
    a = x(4);
    e = x(5);
    M0 = x(6);

    J2 = 1082.63*10^(-6);
    r_eq = 6371; % km

    n = sqrt(mu/a^3);
    p = a*(1-e^2);

    dOdt = -3/2*J2*n * (r_eq/p)^2 * cosd(i) * 180/pi;
    didt = 0;
    dodt = 3/4*J2*n * (r_eq/p)^2 * (5*cosd(i)^2 - 1) * 180/pi;
    dadt = 0;
    dedt = 0;
    dM0dt = 3/4*J2*n * (r_eq/p)^2 * sqrt(1-e^2) * (3*cosd(i)^2 - 1);

    deboldt = [dOdt;didt;dodt;dadt;dedt;dM0dt];

end
```