ASE 389P.4 Methods of Orbit Determination
Homework 1: Basic Orbit Propagation

Junette Hsin

Masters Student, Aerospace Engineering and Engineering Mechanics, University of Texas, Austin, TX 78712

This assignment is designed to provide a basic introduction on how to propagate an orbit. MATLAB was used to complete the assignment.

I. Introduction

In this assignment, a tool was created to numerically propagate a circular orbit about the Earth and to convert between Cartesian and Keplerian orbital elements. 2 solutions for the derivation of the gradient potential function are given. 2-body equations of motion and constants of motion are also explored. The software implemented in this assignment will be used later in the course provides additional background and/or a review of basic orbital mechanics.

II. Problem 1

A. Statement

Given the Earth orbiting spacecraft position and velocity vectors in Cartesian coordinates

$$\underline{R} = -2436.45\hat{i} - 2436.45\hat{j} + 6891.037\hat{k}\ km \tag{1}$$

$$\underline{V} = \dot{\underline{R}} = 5.088611\hat{i} - 5.088611\hat{j} + 0.0\hat{k}\ km/s \tag{2}$$

solve for the Keplerian elements (a, e, i, Ω, ω, ν). Provide your values in the write-up. See the lecture notes

Assume $\mu = 398600.5$ km^3/s^2.

B. Solution

The algorithms to convert orbital elements to Cartesian and back were taken from References [1] and [2]. First, the specific angular momentum vector, h, and perpendicular node vector (to the plane of the orbit), n, were calculated. Inclination, eccentricity, and the rest of the orbital elements followed while checking for equatorial orbits, NaNs, and quadrants of angles. Given the spacecraft position and velocity vectors from the problem statement, the Keplerian elements are:

$$\begin{aligned} a &= 7.712184983762814e+03 \\ e &= 0.447229247404423 \\ i &= 1.570796326794897 \\ \omega &= 3.139593866862924 \\ \Omega &= 3.926990816987241 \\ \nu &= 2.032461649676350 \end{aligned} \tag{1}$$

where a is the semi-major axis, e is the eccentricity, i is the orbit inclination, ω is the argument of perigee, Ω is the right ascension of the ascending node, and ν is the true anomaly.

III. Problem 2

A. Statement

Convert the Keplerian elements from Problem 1 back to position and velocity and provide the values in the write-up. See the lecture notes:

B. Solution

The expression for position in the perifocal frame is the following:

$$\underline{R} = rcos(\nu)\hat{P} + rsin(\nu)\hat{Q} \tag{2}$$

where the scalar magnitude r can be determined from the polar equation of a conic:

$$r = \frac{p}{1 + ecos(\nu)} \tag{3}$$

where

$$p = a(1 - e^2) = \frac{h^2}{\mu} \tag{4}$$

$\underline{\text{R}}$ then needs to be transformed from the perifocal frame to the ECI frame, which can be done through a series of transformation matrices as outlined in References [1] and [3]. The resulting position and velocity vectors returned are:

$$\begin{aligned} r &= [-2.416951611809028e + 03, 2.416951611809029e + 03, -6.904756183437363e + 03] \\ v &= [-5.088570340542163, 5.088570340542164, -0.028767991782603] \end{aligned} \tag{5}$$

IV. Problem 3

A. Statement

Given the gravity potential function $U = \mu/R$, solve for the two-body acceleration due to gravity, i.e.,

$$\nabla U = \frac{\partial U}{\partial x}\hat{i} + \frac{\partial U}{\partial y}\hat{j} + \frac{\partial U}{\partial z}\hat{k} \tag{3}$$

where $R = \underline{R} \bullet \underline{R}$. Include your derivation in your solution to the assignment.

B. Solution 1

To solve for the two-body acceleration due to gravity, first calculate R, which is given in the problem statement as $R = \underline{R} \cdot \underline{R}$. Let us first define $\underline{\text{R}}$ as the following:

$$\underline{R} = x\hat{i} = y\hat{j} + z\hat{k} \tag{6}$$

The dot product of the vector $\underline{\text{R}}$ with itself is equal to the square of its magnitude:

$$R = \underline{R} \cdot \underline{R} = x^2 + y^2 + z^2 \tag{7}$$

Given the gravity potential function $U = \mu/R$, the gradient of the gravity potential function ∇U is:

$$\nabla U = \frac{\delta U}{\delta x}\hat{i} + \frac{\delta U}{\delta y}\hat{j} + \frac{\delta U}{\delta z}\hat{k} \tag{8}$$

First derive $\frac{\delta U}{\delta x}\hat{i}$:

$$\frac{\delta U}{\delta x}\hat{\imath} = \frac{\delta\left(\frac{\mu}{R}\right)}{\delta x}\hat{\imath} = \frac{\delta\left(\frac{\mu}{x^2+y^2+z^2}\right)}{\delta x}\hat{\imath} \tag{9}$$

Take the partial derivative:

$$\frac{\delta\left(\frac{\mu}{x^2+y^2+z^2}\right)}{\delta x}\hat{\imath} = \frac{\delta\left(\mu(x^2+y^2+z^2)^{-1}\right)}{\delta x}\hat{\imath} = -\mu(x^2+y^2+z^2)^{-2}(2x)\hat{\imath} \tag{10}$$

Now simplify:

$$-\mu(x^2+y^2+z^2)^{-2}(2x)\hat{\imath} = -\frac{2\mu x}{(x^2+y^2+z^2)^2}\hat{\imath} \tag{11}$$

Thus:

$$\frac{\delta U}{\delta x}\hat{\imath} = -\frac{2\mu x}{(x^2+y^2+z^2)^2}\hat{\imath} \tag{12}$$

$\frac{\delta U}{\delta y}\hat{\jmath}$ and $\frac{\delta U}{\delta z}\hat{k}$ can be derived through the same process, which result in the following:

$$\frac{\delta U}{\delta y}\hat{\jmath} = -\frac{2\mu y}{(x^2+y^2+z^2)^2}\hat{\jmath} \tag{13}$$

$$\frac{\delta U}{\delta z}\hat{k} = -\frac{2\mu z}{(x^2+y^2+z^2)^2}\hat{k} \tag{14}$$

The gradient of the gravity potential function is thus:

$$\nabla U = -\frac{2\mu x}{(x^2+y^2+z^2)^2}\hat{\imath} - \frac{2\mu y}{(x^2+y^2+z^2)^2}\hat{\jmath} - \frac{2\mu z}{(x^2+y^2+z^2)^2}\hat{k} \tag{15}$$

Plug in $x = -2436.45$, $y = -2436.45$, $z = 6891.037$, and $\mu = 398600.5$ from Problem 1 into Equation 15 to solve for the two-body acceleration due to gravity:

$$\nabla U = 5.512551407304731e - 07\hat{\imath} - 5.512551407304731e - 07\hat{\jmath} - -1.559120676071291e - 06\hat{k} \tag{16}$$

C. Solution 2

There may have been a typo with the gravitational potential function in the problem statement. The gravitational potential function is commonly simplified as $U = \mu/r$, where r is the distance between two bodies [1]:

$$r = \sqrt{R} = \sqrt{\underline{R}\cdot\underline{R}} = (\underline{R}\cdot\underline{R})^{1/2} = (x^2+y^2+z^2)^{1/2} \tag{17}$$

If we use Equation 17 to calculate the gradient for the potential function instead of Equaton 7, then $\frac{\delta U}{\delta x}\hat{\imath}$ becomes:

$$\frac{\delta U}{\delta x}\hat{\imath} = \frac{\delta\left(\frac{\mu}{r}\right)}{\delta x}\hat{\imath} = \frac{\delta\left(\frac{\mu}{(x^2+y^2+z^2)^{1/2}}\right)}{\delta x}\hat{\imath} = \frac{\delta\left(\mu(x^2+y^2+z^2)^{-1/2}\right)}{\delta x} \tag{18}$$

Take the derivative and simplify:

$$\frac{\delta\left(\mu(x^2+y^2+z^2)^{-1/2}\right)}{\delta x} = \mu\left(-\frac{1}{2}\right)(x^2+y^2+z^2)^{-3/2}(2x)\hat{\imath} = -\frac{\mu x}{(x^2+y^2+z^2)^{3/2}}\hat{\imath} \tag{19}$$

Thus:

$$\frac{\delta U}{\delta x}\hat{\imath} = -\frac{\mu x}{(x^2+y^2+z^2)^{3/2}}\hat{\imath} = -\frac{\mu x}{r^3}\hat{\imath} \tag{20}$$

And:

$$\frac{\delta U}{\delta y}\hat{j} = -\frac{\mu y}{r^3}\hat{j} \tag{21}$$

$$\frac{\delta U}{\delta z}\hat{k} = -\frac{\mu z}{r^3}\hat{k} \tag{22}$$

Equation 15 then becomes:

$$\nabla U = -\frac{\mu x}{r^3}\hat{i} - \frac{\mu y}{r^3}\hat{j} - \frac{\mu z}{r^3}\hat{k} \tag{23}$$

Plug in $x = -2436.45$, $y = -2436.45$, $z = 6891.037$, and $\mu = 398600.5$ from Problem 1 into Equation 23 to solve for the two-body acceleration due to gravity:

$$\nabla U = 0.002123566317530\hat{i} + 0.002123566317530\hat{j} - 0.006006104810708\hat{k} \tag{24}$$

V. Problem 4

A. Statement

Develop the necessary code to numerically integrate the equations of motion using the position and velocity from Problem 1 as the initial conditions. Compute the future position and velocity at 20-second intervals for two full orbits. Plot the magnitude of the position, velocity, and acceleration as a function of time for two full orbits and provide the figure. Compute the specific orbital angular momentum vector for these two full orbits and plot that as well, as a function of time, as a 3D scatter plot ($\underline{h} = \underline{R}\ X\ \underline{V}$). Assume that the motion is only due to the accelerations derived from Eq(3)

B. Solution

The period for the orbit was found by using the semi-major axis in the following equation:

$$T = \left|2\pi\sqrt{a^3/\mu}\right| \tag{25}$$

The orbit was propagated by numerically integrating the equations of motion 23 for 2 periods using `ode45`. The relative and absolute tolerances were set to 1e-8. The acceleration was calculated by differentiating velocity with respect to time, and then the magnitudes of the position, velocity, and acceleration were plotted in Figure 1.

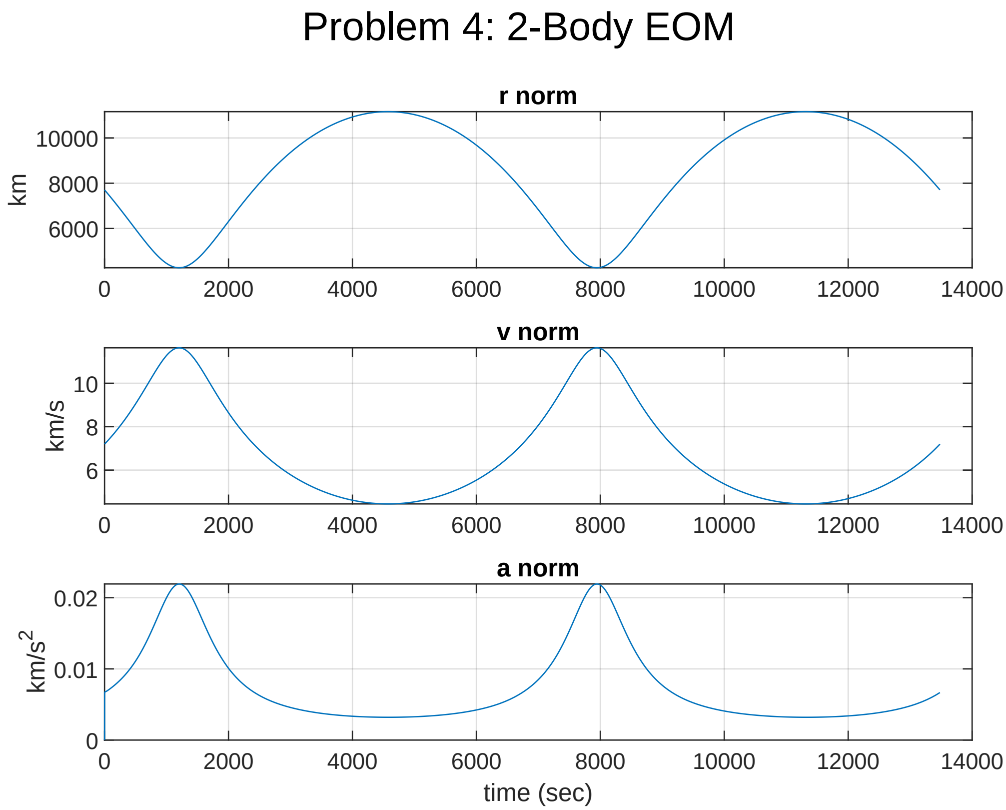

Fig. 1

The magnitudes of the velocity and acceleration vary throughout the orbit, which indicates that the orbit is not circular. The position and velocity vectors were converted into orbital elements which are shown in 2. All of the orbital elements with the exception of true anomaly remain essentially constant, revealing the predictable and Keplerian nature of the orbit. The eccentricity shows that the orbit is elliptical; an eccentricity of 0 forms a perfectly circular orbit, a 1 forms a parabolic escape orbit, and greater than 1 forms a hyperbolic orbit.

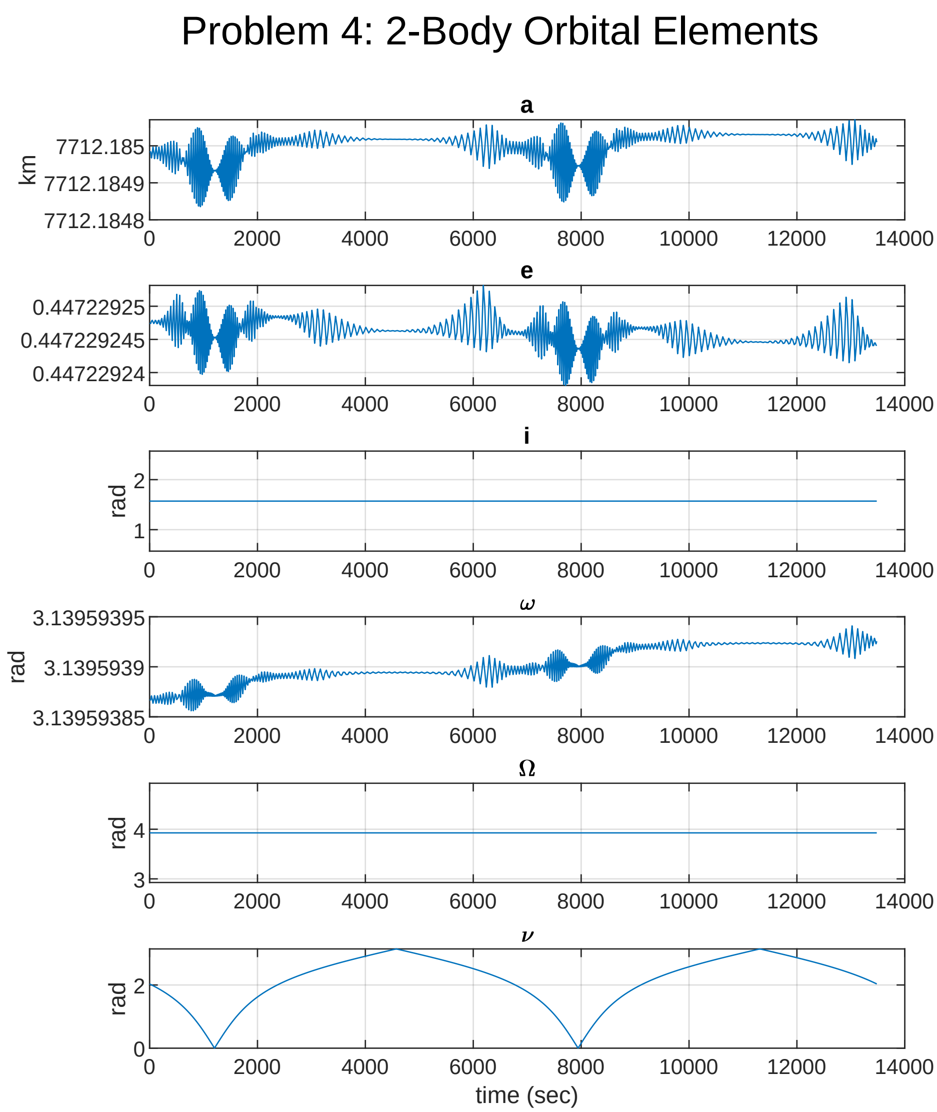

Fig. 2

Figure 3 illustrates that the specific angular momentum essentially remains constant throughout the entire orbit. The difference between the maximum and minimum of **h norm** is 7.204597714007832e-04, which is incredibly small especially when considering that the order of magnitude for all values of **h norm** is 4.

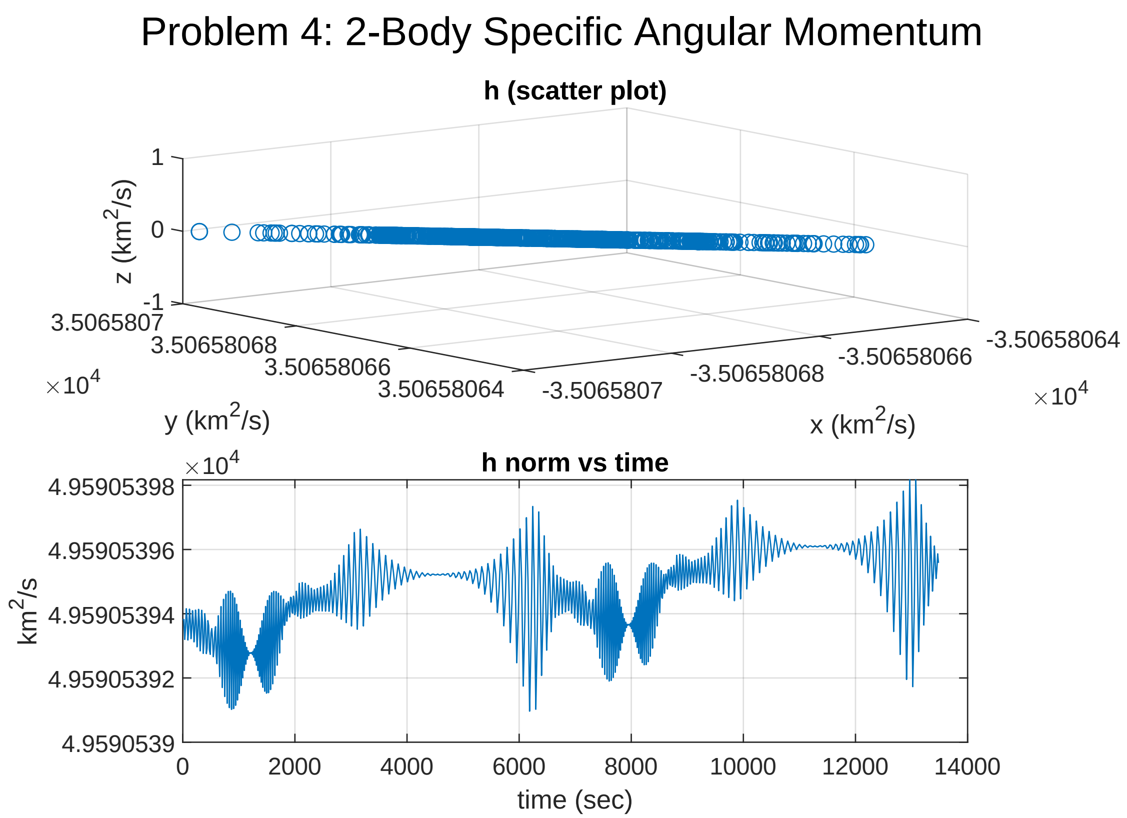

Fig. 3

VI. Problem 5

A. Statement

Compute the specific kinetic energy and specific potential energy as a function of time and plot the change in total specific energy to show that it remains constant over the two orbits. (i.e. plot $dE = E(t) - E(t_0)$). Include the image in your write-up. Why is the change in total specific energy not constant?

B. Solution

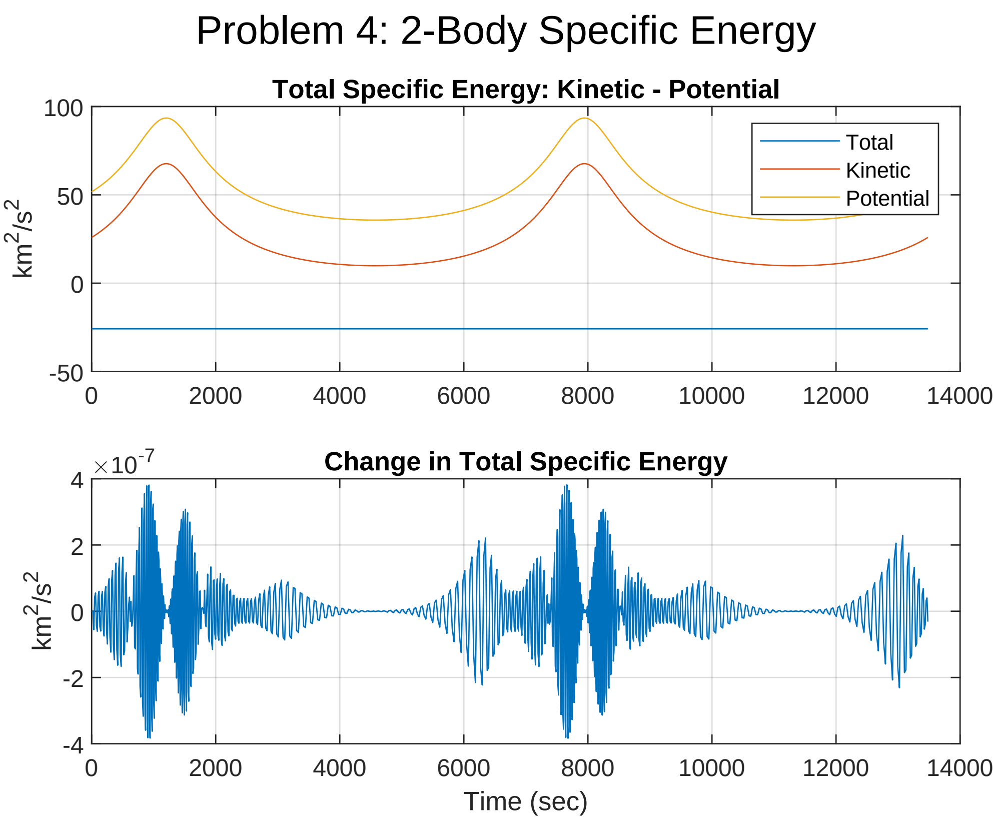

Fig. 4

The energy constant of motion is given in Reference [1]:

$$E = \frac{v^2}{2} - \frac{\mu}{r} \tag{26}$$

in which the first term is the "kinetic energy per unit mass," or specific kinetic energy, and the last term is the specific potential energy. The potential energy will always be negative due to setting the zero reference of potential energy at the center of a massive body and the work done when moving from one point in space to another against the force of gravity. The specific mechanical energy, E, is the sum of the specific kinetic and potential energy and remains constant along its orbit.

VII. Conclusion

Calculating orbital elements to and from Cartesian position and velocity vectors can be tricky because of the quadrant ambiguities when performing trigonometric computations. Problem 3 might have included a typo in its problem statement for the gravity potential function; the potential is commonly given as $U = \mu/r$ in which r is the distance between 2 bodies, not $U = \mu/(\underline{R} \cdot \underline{R})$. Solutions were given for both potential equations. Though position and velocity may vary along an orbit, there are constants of motion such as angular momentum and specific mechanical energy which govern the motion of celestial bodies.

VIII. Appendix

A. HW1 MATLAB code

```
% ASE 389 Orbit Determination
% HW 1
% Junette Hsin

%% Problem 1

global mu

mu = 398600.5 ;
r  = [ -2436.45; -2436.45; 6891.037 ];
v  = [ 5.088611; 5.088611; 0 ];

rv = [r; v];
oe = rv2oe(rv);

%% Problem 2

rv = oe2rv(oe);

%% Problem 3

x = -2436.45;
y = -2436.45;
z = 6891.037;
mu = 398600.5;

dux = -2*mu*x / ( x^2 + y^2 + z^2 )^2;
duy = -2*mu*y / ( x^2 + y^2 + z^2 )^2;
duz = -2*mu*z / ( x^2 + y^2 + z^2 )^2;

rnorm = sqrt( x^2 + y^2 + z^2 );

dux = -mu*x / ( rnorm )^3;
duy = -mu*y / ( rnorm )^3;
duz = -mu*z / ( rnorm )^3;

%% Problem 4

a = oe(1);
T   = abs(2 * pi * sqrt(a^3 / mu));         % period

toler   = 1e-8;          % 1e-14 accurate; 1e-6 coarse
options = odeset('reltol', toler, 'abstol', toler );
[t,x] = ode45(@TwoBod_6states, [0 2*T], [r; v], options);

for i = 1:length(t)
rnorm(i) = norm(x(i, 1:3));
vnorm(i) = norm(x(i, 4:6));
H(i, :) = cross(x(i, 1:3), x(i, 4:6));
hnorm(i) = norm(H(i, :));
end

anorm = 0;
for i = 2:length(t)
a = (x(i, 4:6) - x(i-1, 4:6)) / ( t(i) - t(i-1) );
anorm(i) = norm(a);
end

% ------------------------------------------------------------------

name = 'Problem 4: 2-Body EOM';
h = figure('name', name);

% position
```

```
subplot(3,1,1)
plot(t, rnorm); grid on
title('r norm')
ylabel('km')

% velocity
subplot(3,1,2)
plot(t, vnorm); grid on
title('v norm')
ylabel('km/s')

% acceleration
subplot(3,1,3)
plot(t, anorm); grid on
title('a norm');
ylabel('km/s^2')
xlabel('time (sec)')

sgtitle(name)

save_pdf(h, 'prob4_2bodeom');

% -----------------------------------------------------------------------

name = 'Problem 4: 2-Body EOM Orbit';
h = figure('name', name);
plot3(x(:,1), x(:,2), x(:,3)); hold on; grid on;
plot3(x(1,1), x(1,2), x(1,3), 'o')
plot3(x(end,1), x(end,2), x(end,3), 'x')
xlabel('x (km)')
ylabel('y (km)')
zlabel('z (km)')
legend('orbit', 'start', 'end')

sgtitle(name)

save_pdf(h, 'prob4_2bodeom_orbit');

% -----------------------------------------------------------------------

clear oe
for i = 1:length(t)
oe(i,:) = rv2oe(x(i,:));
end

labels = {'a', 'e', 'i', '\omega', '\Omega', '\nu'};
units = {'km', '', 'rad', 'rad', 'rad', 'rad'};
name = 'Problem 4: 2-Body Orbital Elements';
h = figure('name', name, 'position', [100 100 500 600]);
for i = 1:6
subplot(6,1,i)
plot(t, oe(:, i)); grid on
title(labels{i});
ylabel(units{i});
end
xlabel('time (sec)')
sgtitle(name)

save_pdf(h, 'prob4_2bodoes');

% -----------------------------------------------------------------------

name = 'Problem 4: 2-Body Specific Angular Momentum';
h = figure('name', name);
subplot(2,1,1)
scatter3(H(:,1), H(:,2), H(:,3)); grid on
xlabel('x (km^2/s)')
ylabel('y (km^2/s)')
```

```
zlabel('z (km^2/s)')
title('h (scatter plot)')

subplot(2,1,2)
plot(t, hnorm); grid on
xlabel('time (sec)')
ylabel('km^2/s')
title('h norm vs time')

sgtitle(name)

save_pdf(h, 'prob4_angmom')

%% Problem 5

% specific kinetic energy
for i = 1:length(t)
T(i) = 0.5 * vnorm(i)^2;
U(i) = mu / rnorm(i);
end
E = T - U;

name = 'Problem 4: 2-Body Specific Energy';
h = figure('name', name);
subplot(2,1,1)
plot(t, E); grid on; hold on;
plot(t, T);
plot(t, U);
ylabel('km^2/s^2')
legend('Total', 'Kinetic', 'Potential')
title('Total Specific Energy: Kinetic - Potential')
subplot(2,1,2)
plot(t, [0 diff(E)]); grid on
title('Change in Total Specific Energy')
xlabel('Time (sec)')
ylabel('km^2/s^2')
sgtitle(name)

save_pdf(h, 'prob5_energy')

%% subfunctions

function save_pdf(h, name)

% save as cropped pdf
set(h,'Units','Inches');
pos = get(h,'Position');
set(h,'PaperPositionMode','Auto','PaperUnits','Inches','PaperSize',[pos(3), pos(4)])
print(h,name,'-dpdf','-r0')

end
```

B. rv2oe function

```
function oe = rv2oe(rv)
% ---------------------------------------------------------------------
% Inputs
%   rv = [6x1] position and velocity states vector
%
% Outputs
%   oe = [6x1] orbital elements: a, e, i, w, Omega, nu
%           a       = semimajor axis
%           e       = eccentricity
%           i       = inclination
%           w       = argument of perigee
%           Omega   = right ascension of ascending node
%           nu      = true anomaly
```

```
% ----------------------------------------------------------------------------

global mu

r = rv(1:3);
v = rv(4:6);

% angular momentum
h       = cross(r,v);

% node vector
nhat    = cross( [0 0 1], h );

% eccentricity
evec    = ( (norm(v)^2 - mu/norm(r))*r - dot(r,v)*v ) / mu;
e       = norm(evec);

% specific mechanical energy
energy  = norm(v)^2/2 - mu/norm(r);

% semi-major axis and p
if abs(e-1.0)>eps
a = -mu/(2*energy);
p = a*(1-e^2);
else
p = norm(h)^2/mu;
a = inf;
end

% inclination
i = acos(h(3)/norm(h));

% right ascension of ascending node (check for equatorial orbit)
if i > 0.000001
Omega = acos( nhat(1)/norm(nhat) );
else
Omega = 0;
end
if isnan(Omega)
Omega = 0;
end
if nhat(2)<0
Omega = 2*pi - Omega;
end

% argument of perigee
if e > 0.000001
w = acos(dot(nhat,evec)/(norm(nhat)*e));
else
w = 0;
end
if isnan(w)
w = 0;
end
% if e(3)<0
%     argp = 360-argp
% end

% true anomaly
nu = acos( dot(evec,r) / (e*norm(r)) );
% if dot(r,v)<0
%     nu = 360 - nu
% end

oe = [a; e; i; w; Omega; nu];

end
```

C. oe2rv function

```
function [rv] = oe2rv(oe)
% ---------------------------------------------------------------------------
% Purpose: Convert orbital elements and time past epoch to the classic
% Cartesian position and velocity
%
% Inputs:
%   oe       = [6x1] or [1x6] orbital elements
%   delta_t  = t - t0 time interval
%   mu       = Gravity * Mass (of Earth) constant
%
% Outputs:
%   rv       = position and velocity state vector
% ---------------------------------------------------------------------------

% global delta_t
global mu

a        = oe(1);
e        = oe(2);
i        = oe(3);
w        = oe(4);
LAN      = oe(5);
% M0       = oe(6);
nu       = oe(6);

% nu is TRUE ANOMALY --> use Kepler's to calculate MEAN ANOMALY
% E = 2*atan( sqrt( (1-e)/(1+e) ) * tan(nu/2) );
% M = M0 + sqrt( mu/a^3 ) * (delta_t);
% E = keplerEq(M, e, eps);
% E = kepler(M, e);
% nu = 2*atan( sqrt( (1+e)/(1-e) ) * tan(E/2) );

p = a * ( 1 - e^2 );              % intermediate variable
r = p / ( 1 + e*cos(nu) );        % r_magnitude, polar coordinates

% Perifocal position and velocity

r_pf = [ r * cos(nu); r * sin(nu); 0 ];
v_pf = [ -sqrt(mu/p) * sin(nu); sqrt(mu/p) * (e + cos(nu)); 0 ];

% Perifocal to ECI transformation, 3-1-3 rotation
R11 = cos(LAN)*cos(w) - sin(LAN)*sin(w)*cos(i);
R12 = -cos(LAN)*sin(w) - sin(LAN)*cos(w)*cos(i);
R13 = sin(LAN)*sin(i);

R21 = sin(LAN)*cos(w) + cos(LAN)*sin(w)*cos(i);
R22 = -sin(LAN)*sin(w) + cos(LAN)*cos(w)*cos(i);
R23 = -cos(LAN)*sin(i);

R31 = sin(w)*sin(i);
R32 = cos(w)*sin(i);
R33 = cos(i);

R = [R11 R12 R13; R21 R22 R23; R31 R32 R33];

% Transform perifocal to ECI frame
r_vec = R * r_pf;
v_vec = R * v_pf;

% Position and state vector
rv = [r_vec; v_vec];

end

%% Kepler equation solvers

```

```
function E = keplerEq(M,e,eps)
% Function solves Kepler's equation M = E-e*sin(E)
% Input - Mean anomaly M [rad] , Eccentricity e and Epsilon
% Output  eccentric anomaly E [rad].
En  = M;
Ens = En - (En-e*sin(En)- M)/(1 - e*cos(En));
while ( abs(Ens-En) > eps )
En = Ens;
Ens = En - (En - e*sin(En) - M)/(1 - e*cos(En));
end
E = Ens;
end

function E = kepler(M, e)
f = @(E) E - e * sin(E) - M;
E = fzero(f, M);  % <-- I would use M as the initial guess instead of 0
end
```

s

References

[1] Donald D. Mueller, J. W., and Bate, R. R., *Fundamentals of Astrodynamics*, Dover Publications, Inc., 1971.

[2] Jah, M. K., "ASE 389P.4 Methods of Orbit Determination Module 3," , January 2021.

[3] Jah, M. K., "ASE 389P.4 Methods of Orbit Determination Module 2," , January 2021.