1. Calculate the (classical) Keplerian orbital elements at epoch (present angles in degrees).

a=6891745.73679017 e=0.00261951184587647 i=92.6714363035991
ω=-159.827438612996 Ω=144.207168070987 ν= -10.8432571800292

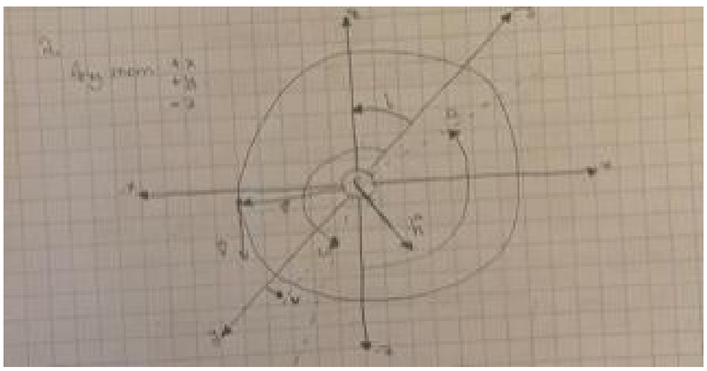
2.. Draw a neat sketch showing the orbit (obviously, not to scale), and clearly draw or label the following:

a. The inertial XYZ axes

b. The radius vector and the velocity vector of the satellite at a location in orbit where the argument of latitude is approximately 30°.

c. The six Keplerian orbital elements

d. The angular momentum vector.



3. Calculate the mean angular velocity of GRACE-FO (show your work, give units).

From formula of mean motion $n = \sqrt{\mu/a^3}$ (found in Vallardo text, pg 31)
meanAngVel =0.001103506811686

a. Calculate the instantaneous angular velocity of GRACE-FO at the given epoch. (show your work, give units)

From formula *v=rw* so = v/r
instantAngVel = 0.001109203742687

b. Explain why the mean and instantaneous velocities are different. (try to connect your answer back to the balance between inertia and gravity – do not simply cite different formulas for the two)

The instantaneous velocity will have minute variations at different instances due to the eccentricity making the orbit *near* circular but not perfect.

4. Calculate the Cartesian position and velocity at epoch, starting from orbital element values calculated in (1).

a. Calculate the difference between the original and reconstructed position and velocity.

Difference:
rx = -9.31322574615479e-10
ry = -4.65661287307739e-10
rz = -1.86264514923096e-09
Magnitude of position difference = 2.1339e-09
vx = 0
vy = 1.42108547152020e-12
vz = 9.09494701772928e-13
Magnitude of velocity difference = 1.6872e-12

b. If these differences are not exactly zero, write a few sentences discussing why.

Both of the differences are of such low magnitude that they are quite nearly zero, so the only explanation for not being exactly zero is computational or processing error.

5. Calculate the specific energy and specific angular momentum vector. Give units.

specificEnergy = -2.8919e+07 m^2/s^2
specificAngMom = [30620260917.4414 42467236165.1173 -2442850259.53202]
Magnitude Angular Momentum = 5.2412e+10 m^2/s

6. Using a numerical integrator, assuming simple two-body inverse-square law acceleration, propagate the GRACE-FO orbit forward for one quarter of an orbital period.

a. Compare the numerically integrated position and velocity with analytically propagated position and velocity, at one-minute intervals– comment on the differences, if any. [Analytical propagation of the orbit may be done using the method labeled "Orbital Elements" in Section 2.3.1 of the Vallado text].

- Find the orbital period and then the quarter of the orbital period

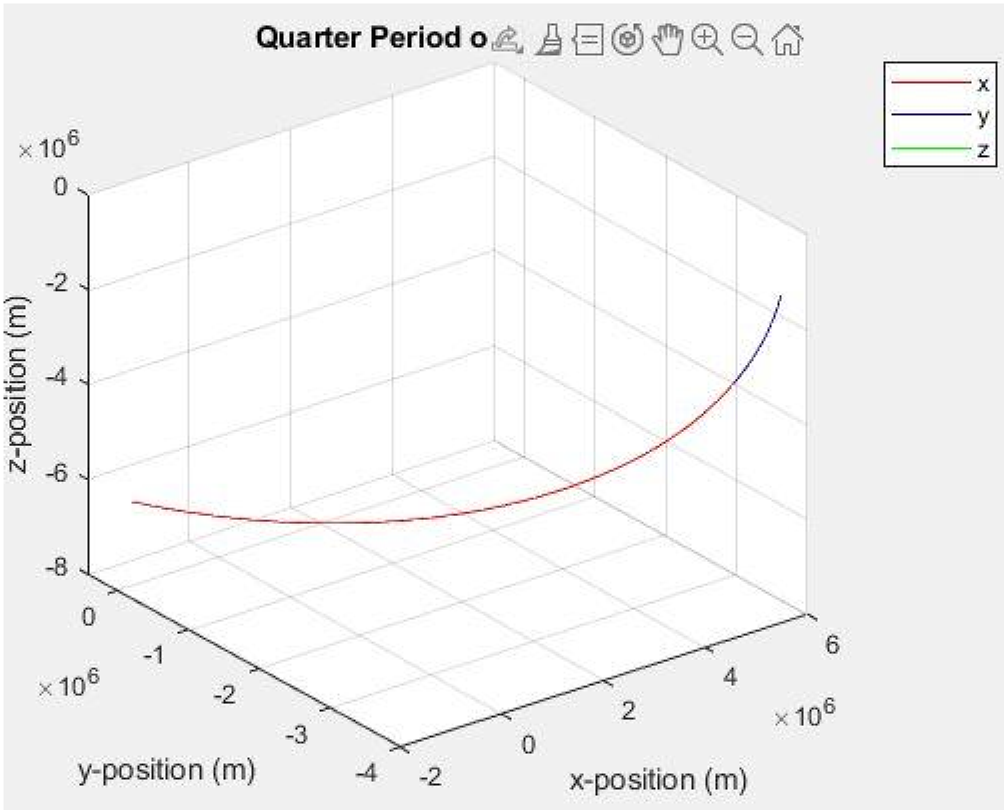
  $T = 2\pi\sqrt{a^3/\mu}$ = 5.6938e+03

  $T_{1/4}$ = 1.4235e+03
- Use 2 Body equation of motion and ODE45 Differential equation solver with the equation shown

  $\bar{F} = (\mu m/r^3) * \bar{r}$
- In order to plot the analytically propagated position and velocity the equations for mean motion(n), time of flight (TOF), eccentric anomaly (E), and mean anomaly (M) were used.

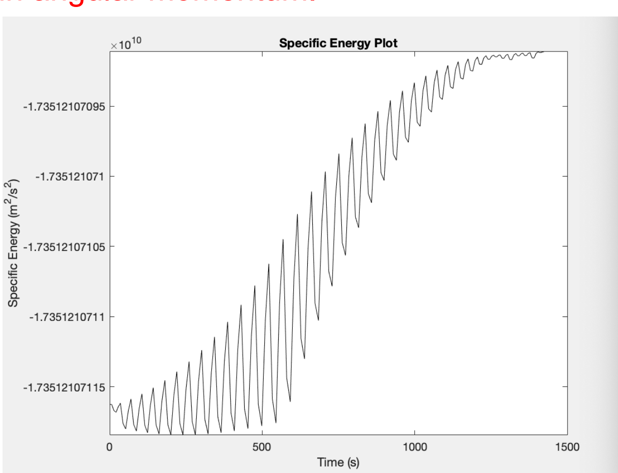
  E1 = -0.3775
  M* = -5.089
  E2= -5.087
  v2 = 1.199

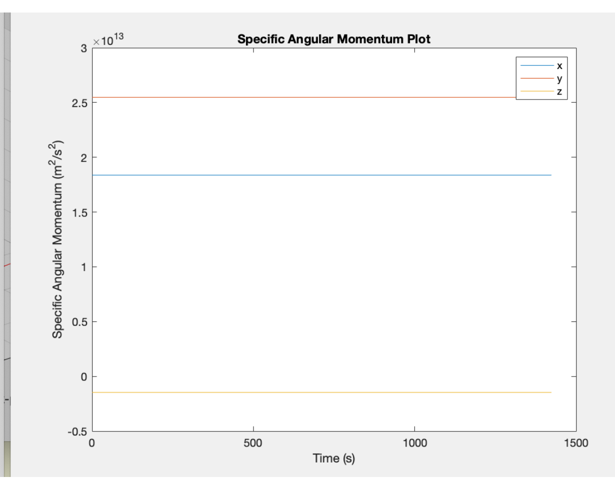
- The plot can now be made by substituting the true anomaly and linearly spacing it with the original true anomaly and converting it to rv vectors. It can be seen in the plot that there is essentially no difference between the trajectories.



b. At each step, re-calculate the energy and angular momentum vector, and check if (in your opinion) these remain constant.

I found these to remain constant along the same reasoning. With essentially no variation in energy when magnitude is taken into account, and no visible change in angular momentum.





c. Write a comment on why you believe your choice of the numerical integrator was adequate for the purposes of answering this problem.

From my coding work in 366k, ODE45 was the MATLAB tool recommended by Professor Russell. It worked well throughout the course, so I believe it to be adequate for accurately propagating the position and velocity in this problem. It also seems to be the most user friendly differential equation solver MATLAB offers.

7. A second pair of inertial axes X1Y1Z1 are obtained from the ECI through two rotations. The first rotation is about the Z-axis by 144°; and the second rotation is around the intermediate Y-axis by +23.5°.

a. Calculate the orbital elements relative to the X1Y1Z1. Write all expressions used.

```
R3 = [cosd(theta) sind(theta) 0;
      -sind(theta) cosd(theta) 0;
           0           0       1];

R2 = [cosd(theta) 0 sind(theta);
        0          1      0    ;
     sind(theta)   0  cosd(theta)];
```

$r_{rot} = R_2(23.5°)R_3(144°)r_{ECI}$

$v_{rot} = R_2(23.5°)R_3(144°)v_{ECI}$

a = 6.8917e+06 m

e = 0.0026

i = 92.3671 degrees

ω = 1.2558 degrees

Ω = -136.3063 degrees

ν = -10.8433 degrees

b. Which elements remain unchanged compared to the elements calculated in (1)? Why?

The 3 elements to change were inclination, argument of periapse, and longitude of ascending nodes. Let it be noted that inclination only changed slightly. The semimajor axis, eccentricity, and true anomaly are all related to scalars (i.e the position and velocity) of the spacecraft or orbiter with respect to the orbit in some way and are not impacted by rotation of the axes.

Code Appendix:

```
%% Applied Orbital Mechanics HW#1

%constants
mu = 3.986004415*10^14;
ae = 6378136.3;
we = 7.292115*10^-5;

% Satellite: GRACE Follow-On (google it, if you want more information)
%
% Epoch: Specified in UTC Time System
%        EPOCH1:  Year    Month    Day
%        EPOCH2:  Hour    Minute   Seconds
%
% POS (Position, ECI, in meters)         X       Y        Z
% VEL (Velocity, ECI, in meters/second)  Xdot    Ydot     Zdot
%
%        EPOCH1          2018.0          11.0            30.0
%        EPOCH2            23.0          59.0            42.0
%        POS        5471639.55639308   -4009260.88949393   -1113125.19797190
%        VEL       -1210.66329250225    440.63844202350   -7515.04479126903

%% PROBLEM 1
% Calculate the (classical) Keplerian orbital elements at epoch (present angles in degrees).
rvGraceFO = [5471639.55639308   -4009260.88949393   -1113125.19797190,
-1210.66329250225    440.63844202350   -7515.04479126903];
oeGFO = hw6rv2oe(rvGraceFO,mu);
oeGraceFOinDegrees = [oeGFO(1:2)' oeGFO(3:6)'.*180./pi]

%% PROBLEM 2

%% PROBLEM 3
instantAngVel=norm(rvGraceFO(4:6))/norm(rvGraceFO(1:3))
meanAngVel=sqrt(mu/oeGFO(1)^3)
  %instantaneous velocity will have minute variations on different instances due to the
  %eccentricity making it near circular but not perfect.

%% PROBLEM 4
rvRecalc = hw6oe2rv(oeGFO,mu);
difference = rvGraceFO' - rvRecalc
magRdiff = norm(difference(1:3))
```

```
magVdiff = norm(difference(4:6))

%% PROBLEM 5
specificEnergy = -mu/(2*oeGFO(1))
specificAngMom = cross(rvGraceFO(1:3), rvGraceFO(4:6))
angMomMag = norm(specificAngMom)

%% PROBLEM 6
%Propagation
T = 2*pi*sqrt(oeGFO(1)^3/mu);
Tquarter = T/4;

E0=2*atan(sqrt((1-oeGFO(2))/(1+oeGFO(2)))*tan(oeGFO(6)));
M=sqrt(mu/(oeGFO(1)^3))*Tquarter-2*pi+E0-oeGFO(2)*sin(E0);
Ef=hw7kepler(M,oeGFO(2));
nuF=2*atan(sqrt((1+oeGFO(2))-(1-oeGFO(2)))*tan(Ef/2));
nuSpaced=linspace(oeGFO(6),nuF,109)';

oe=zeros(109,6);
oe(:,1)=oeGFO(1);
oe(:,2)=oeGFO(2);
oe(:,3)=oeGFO(3);
oe(:,4)=oeGFO(4);
oe(:,5)=oeGFO(5);
oe(:,6)=nuSpaced;

rv = zeros(109,6);
for i=1:109
   rv(i,:)=hw6oe2rv(oe(i,:),mu);
end

options = odeset('RelTol', 1e-9, 'AbsTol', 1e-9);
[t,rv6]=ode45(@(t,r) body2(t,r,mu),[0 Tquarter], rvGraceFO, options);
figure(1)
plot3(rv6(:,1),rv6(:,2),rv6(:,3),"r",rv(:,1),rv(:,2),rv(:,3),"b",0,0,0,"g")
grid
title("Quarter Period of Orbit Plot")
xlabel("x-position (m)")
ylabel("y-position (m)")
zlabel("z-position (m)")
legend("x","y","z")
```

```
%Energy
G=6.67408e-11;
mE=mu/G;
mg=600;
[k,u]=energy(rv,mE,mg,G)
E=k-u;
figure(2)
plot(t,E,"k")
title("Specific Energy Plot")
xlabel("Time (s)")
ylabel("Specific Energy (m^2/s^2)")

%Angular Momentum
h=angmom(rv6,mg);
figure(3)
plot(t,h(:,1),t,h(:,2),t,h(:,3))
title("Specific Angular Momentum Plot")
xlabel("Time (s)")
ylabel("Specific Angular Momentum (m^2/s^2)")
legend("x","y","z")

%% PROBLEM 7
r7=[5471639.55639308   -4009260.88949393   -1113125.19797190];
v7=[-1210.66329250225     440.63844202350   -7515.04479126903];

r = R2(23.5)*R3(144)*r7';
v = R2(23.5)*R3(144)*v7';

rvRot = [r' v'];
oeRot = hw6rv2oe(rvRot,mu);
oeRotinDegrees = [oeRot(1:2)' oeRot(3:6)'*180/pi]

function R = R3(theta)
   R = [cosd(theta),sind(theta),0; -sind(theta), cosd(theta), 0; 0, 0, 1];
end
function R = R2(theta)
   R = [cosd(theta), 0, -sind(theta);0, 1, 0; sind(theta), 0, cosd(theta)];
End

function[rv] = hw6oe2rv(oe, mu)

rMag = oe(1)*(1-oe(2)^2)/(1+oe(2)*cos(oe(6)));
```

```
pHat = [1 0 0];
qHat = [0 1 0];

rPQW = rMag*cos(oe(6))*pHat+rMag*sin(oe(6))*qHat;
RbigW = [cos(oe(5)) -1*sin(oe(5)) 0; sin(oe(5)) cos(oe(5)) 0; 0 0 1];
Ri = [1 0 0;0 cos(oe(3)) sin(-1*oe(3));0 sin(oe(3)) cos(oe(3))];
Rw = [cos(oe(4)) -1*sin(oe(4)) 0; sin(oe(4)) cos(oe(4)) 0; 0 0 1];

rIJK = RbigW*Ri*Rw*rPQW';
rv(1,1)=rIJK(1);
rv(2,1)=rIJK(2);
rv(3,1)=rIJK(3);

p = oe(1)*(1-oe(2)^2);
vPQW = -1*sin(oe(6))*sqrt(mu/p)*pHat+sqrt(mu/p)*(oe(2)+cos(oe(6)))*qHat;
vIJK=RbigW*Ri*Rw*vPQW';
rv(4,1)=vIJK(1);
rv(5,1)=vIJK(2);
rv(6,1)=vIJK(3);

end

function[oe] = hw6rv2oe(rv,mu)
r = [rv(1) rv(2) rv(3)];
rMag = norm(r);

v = [rv(4) rv(5) rv(6)];
vMag = norm(v);

oe(1,1) = -1*mu / (2*(vMag^2/2-mu/rMag));

h = cross(r,v);
e = cross(v,h)/mu-(r/rMag);
oe(2,1) = norm(e);

zHat = [0 0 1];

oe(3,1) = acos(dot(zHat,h/norm(h)));
nHat = cross(zHat,h) / norm(cross(zHat,h));
oe(4,1) = acos(dot(nHat,e)/norm(e));
if e(3)<0
   oe(4,1) = -1*oe(4,1);
```

```
end

oe(5,1) = atan2(nHat(2), nHat(1));

oe(6,1) = acos(dot(r,e)/(rMag*norm(e)));
if dot(r,v)<0
   oe(6,1) = -1*oe(6,1);
end
End

function [k,u] = energy(rv,m0,m1,G)
rE = sqrt(rv(:,1).^2+rv(:,2).^2+rv(:,3).^2);
k=0.5*m1*(rv(:,4).^2+rv(:,5).^2+rv(:,6).^2);
u=G*m0*m1./rE;
end

function h = angmom(r,m)
l=length(r);
h=zeros(l,3);
for i=1:l
   h(i,1:3)=cross(r(i,1:3),m*r(i,4:6));
end
end

function [E] = hw7kepler(M,e)
iter = 0;
eps = 1e-14;
Eo = M;
stepE = (-1)*(M - Eo + e*sin(Eo))/(-1 + e*cos(Eo));

while abs(stepE) > eps
   iter = iter + 1;
stepE = (-1)*(M - Eo + e*sin(Eo))/(-1 + e*cos(Eo));
Eo = Eo + stepE;
fprintf('E=%f on iteration %i\n',Eo, iter)
end

E = Eo;
fprintf('Converged! E=%f after %i iteration(s)',E, iter)

function drdt=body2(t,r,mu)
drdt(1)=r(4);
```

```
drdt(2)=r(5);
drdt(3)=r(6);
drdt(4)=-1*mu*r(1)/(r(1)^2+r(2)^2+r(3)^2)^1.5;
drdt(5)=-1*mu*r(2)/(r(1)^2+r(2)^2+r(3)^2)^1.5;
drdt(6)=-1*mu*r(3)/(r(1)^2+r(2)^2+r(3)^2)^1.5;
drdt=drdt';
end
```