Problem 1:

rp1 = rp2 = rp and a1 < a2

For each orbit at periapsis, the speed is:

and

Since a1 < a2,

Therefore:

Since μ is a constant for both orbits, we can conclude that V2 > V1 at periapsis.

For each orbit at apoapsis,

ra1 < ra2

Therefore, each speed is:

and

Since ra1 < ra2 and a1 < a2,

and

Therefore:

Since μ is a constant for both orbits, we can conclude that V1 > V2 at apoapsis.

Problem 1 can also be solved numerically.

clc;clear;

mu = 1;

rp = 10;

a2 = 20;

a1 = 15;

e1 = 1-(rp/a1);

e2 = 1-(rp/a2);

ra1 = (a1\*(1-e1^2))/(1-e1);

ra2 = (a2\*(1-e2^2))/(1-e2);

vp1 = sqrt(mu)\*sqrt((2/rp)-(1/a1));

vp2 = sqrt(mu)\*sqrt((2/rp)-(1/a2));

va1 = sqrt(mu)\*sqrt((2/ra1)-(1/a1));

va2 = sqrt(mu)\*sqrt((2/ra2)-(1/a2));

fprintf('The speeds at apoapsis are %g and %g for orbits 1 and 2\n',va1,va2);

fprintf('The speeds at periapsis are %g and %g for orbits 1 and 2\n',vp1,vp2);

The speeds at apoapsis are 0.182574 and 0.129099 for orbits 1 and 2

The speeds at periapsis are 0.365148 and 0.387298 for orbits 1 and 2

Problem 2:

Function:

function [a, e, p, h, va, vp] = orbitAltitudes(aa,ap,rb,mu)

%This function calculates properties of a earth orbit with inputs of altitude at apoapsis and periapsis

%Function call: [a, e, p, h, va, vp] = orbitAltitudes(aa,ap,rb,mu)

%

%Input: aa, altitude at apoapsis

%Input: ap, altitude at periapsis

%Input: rb, radius of body

%Input: mu, gravitational parameter

%

%Output: a, semi-major axis

%Output: e, eccentricity

%Output: p, semi-latus rectum

%Output: h, magnitude of the specific angular momentum

%Output: va, velocity at apoapsis

%Output: vp, velocity at periapsis

ra = aa+rb;

rp = ap+rb;

a = (ra+rp)/2;

e = (ra-rp)/(ra+rp);

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

h = sqrt(mu\*p);

vp = sqrt(mu)\*sqrt((2/rp)-(1/a));

va = sqrt(mu)\*sqrt((2/ra)-(1/a));

end

Main:

clc;clear;

aa = 800;

ap = 500;

rb = 6378.145;

mu = 398600;

[a, e, p, h, va, vp] = orbitAltitudes(aa,ap,rb,mu);

fprintf('a = %g km\n',a);

fprintf('e = %g\n',e);

fprintf('p = %g km\n',p);

fprintf('h = %g km^2/s\n',h);

fprintf('Va = %g km/s\n',va);

fprintf('Vp = %g km/s\n',vp);

Output:

a = 7028.15 km

e = 0.0213428

p = 7024.94 km

h = 52916.4 km^2/s

Va = 7.37187 km/s

Vp = 7.69341 km/s

Problem 3:

Orbit B would allow for longer visualization of point Q. The solution to problem 1 proved that an orbit with the same periapsis but a larger apoapsis have a slower speed at apoapsis and a faster speed at periapsis. This is the same as orbit B. The spacecraft would travel slowly over point Q while the spacecraft is nearing apoapsis, and is observing point Q, and would travel faster near periapsis, when point Q cannot be observed.

Problem 4:

Given information:

and

Since Orbit 1 is circular:

Substituting known values into the radius of periapsis equation:

Using the speed at periapsis of Orbit 2 and the vis-viva equation to find rp2:

Rearranging for :

We also know that:

Equating the two previous equations:

Solving for e2:

Problem 5:

Function:

function [nu, e, energy, p, h, rp, ra] = flightPathSpeedRadius(v,r,gamma,mu)

%This functions calculates true anomaly, eccentricity, and total energy given speed, flight path angle, and radius

%Function call: [nu e energy p h rp ra] = flightPathSpeedRadius(v,r,gamma,mu)

%

%Input: v, speed

%Input: r, radius

%Input: gamma, flight path angle

%Input: mu, gravitational parameter

%

%Output: nu, true anomaly

%Output: e, eccentricity

%Output: energy, total mechanical energy

%Output: p, semi-latus rectum

%Output: h, magnitude of the specific angular momentum

%Output: rp, periapsis radius

%Output: ra, apoapsis radius

a = ((((v/sqrt(mu))^2)-(2/r))^-1)\*-1;

h = r\*v\*cos(gamma);

p = (h^2)/mu;

e=sqrt(1-(p/a));

nu = acos((p/(r\*e))-(1/e));

energy = -(mu)/(2\*a);

rp = p/(1+e);

ra = p/(1-e);

end

Main:

clc;clear;

v=7.5;

r=9500;

flightPath=18;

gamma=deg2rad(flightPath);

mu=398600;

[nu, e, energy, p, h, rp, ra] = flightPathSpeedRadius(v,r,gamma,mu);

fprintf('true anomaly = %g rad\n',nu);

fprintf('e = %g\n',e);

fprintf('orbital energy = %g kg km^2 s^-2\n',energy);

true anomaly = 1.07596 rad

e = 0.447706

orbital energy = -13.8329 kg km^2 s^-2

Problem 6:

clc;clear;

rvec = [-12 -20 15];

rvec = (1/20)\*rvec;

mu = 1;

r = norm(rvec);

inertialAccel = -(mu/r^3)\*rvec;

fprintf('Inertial Acceleration in I is: ');

fprintf('%g, %g, %g',inertialAccel(1),inertialAccel(2),inertialAccel(3));

Inertial Acceleration in I is: 0.225088, 0.375146, -0.28136

Problem 7:

From the problem we know that:

and

Using the vis-viva equation to solve for the speed of a circular orbit where r = a:

Solving for r in the vis-viva equation for an ellipse results in:

Substituting v into the previous equation:

We also know from the orbit equation that:

Equating the previous two equations:

Substituting r into the right side and solving for results in:

Problem 8:

Starting with the equation for angular momentum and taking the magnitude:

Know that the zenith angle, and flight path angle, γ, is equal to:

Substituting the equation for zenith angle into the equation for angular momentum:

Taking the scaler product of and :

Since , the scaler product can be rewritten as:

Combining the results leaves:

We also know that:

Substituting this result into the previous equation:

Knowing that and equal:

and

We can substitute for and to result in:

Since the zenith angle is between 0 and π, we can do a substitution to find the range for flight path angle

so therefore

Problem 9:

Function:

function [hVec, eVec, hDote, p, a, nu] = positionVelocity(rVec, vVec, mu)

%This function takes postion and velocity vectors as inputs and calculates orbital quantities

%Function call: [hVec, eVec, hDote, p, a, nu] = positionVelocity(rVec, vVec, mu)

%

%Input: rVec, postion vector

%Input: vVec, velocity vector

%Input: mu, gravitational parameter

%

%Output: hVec, specific angular momentum vector

%Output: eVec, eccentricity vector

%Output: hDote, dot product of specific angular momentum vector and eccentricity vector

%Output: p, semi-latus rectum

%Output: a, semi-major axis

%Output: nu, true anomaly

hVec = cross(rVec,vVec);

r = norm(rVec);

eVec = (cross(vVec,hVec)/mu)-(rVec/r);

hDote = dot(hVec,eVec);

h = norm(hVec);

p = h^2/mu;

e = norm(eVec);

a = abs(p/(1-e^2));

nu = acos((p/(r\*e))-(1/e));

end

Main

clc;clear;

rVec = [0 2 0];

vVec = [(-1/sqrt(3)) (sqrt(2)/sqrt(3)) 0];

mu = 1;

[hVec, eVec, hDote, p, a, nu] = positionVelocity(rVec, vVec, mu);

fprintf('The specific angular momentum vector is: ');

fprintf('%g, %g, %g\n',hVec(1),hVec(2),hVec(3));

fprintf('The eccentricity vector is: ');

fprintf('%g, %g, %g\n',eVec(1),eVec(2),eVec(3));

fprintf('h in I dotted with the eccentricity vector = %g\n',hDote);

fprintf('p = %g km\n',p);

fprintf('a = %g km\n',a);

fprintf('true anomaly = %g rad\n',nu);

The specific angular momentum vector is: 0, -0, 1.1547

The eccentricity vector is: 0.942809, -0.333333, 0

h in I dotted with the eccentricity vector = 0

p = 1.33333 km

a = 3.0024e+15 km

true anomaly = 1.91063 rad

Problem 10:

The orbit equation with semi-latus rectum substituted in is equivalent to:

Since a is equal to r:

Then solving for :

Speed is given by the vis-viva equation:

Since r is equal to a:

Problem 11:

Function:

function [h, p, a, rp, ra]=energyEccentricity(energy,e,mu)

%This function takes orbital energy and eccentricity to calculate h, p, a, rp, ra

%Function Call: [h, p, a, rp, ra]=energyEccentricity(energy,e,mu)

%

%Input: energy, orbital energy

%Input: e, eccentricity

%Input: mu, gravitational parameter

%

%Output: h, magnitude of the specific angular momentum

%Output: p, semi-latus rectum

%Output: a, semi-major axis

%Output: rp, periapsis radius

%Output: ra, apoapsis radius

a = -mu/(2\*energy);

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

h = sqrt(mu\*p);

rp = a\*(1-e);

ra = a\*(1+e);

end

Main:

clc;clear;

givenEnergy = -2\*10^8; %ft^2/s

e = 0.2;

mu = 398600;

energy = givenEnergy / 10763910.41671; %ft^2/s^2 to km^2/s^2

[h, p, a, rp, ra]=energyEccentricity(energy,e,mu);

fprintf('h = %g m^2/s\n',h);

fprintf('p = %g km\n',p);

fprintf('a = %g km\n',rp);

fprintf('rp = %g km\n',rp);

fprintf('ra = %g km\n',ra);

h = 64066.1 m^2/s

p = 10297.2 km

a = 8580.99 km

rp = 8580.99 km

ra = 12871.5 km

Problem 12:

Function:

function [aa, energy, h, p] = periapsisAltEccentricity(ap,e,rb,mu)

%This function takes altitude at periapsis and eccentricity to calculate altitude at apoapsis, orbital energy, h, and p

%Function Call: [aa, energy, h, p] = periapsisAltEccentricity(ap,e,rb,mu)

%

%Input: ap, altitude at periapsis

%Input: e, eccentricity

%Input: rb, radius of body

%Input: mu, gravitational parameter

%

%Output: aa, apoapsis altitude

%Output: energy, orbital energy

%Output: h, magnitude of the specific angular momentum

%Output: p, semi-latus rectum

rp = ap + rb;

a = rp/(1-e);

ra = a\*(1+e);

aa = ra - rb;

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

h = sqrt(mu\*p);

energy = -(mu)/(2\*a);

end

Main:

clc;clear;

e = 0.1;

ap = 370;

rb = 6378.145;

mu = 398600;

[aa, energy, h, p] = periapsisAltEccentricity(ap,e,rb,mu);

fprintf('apoapsis altitude = %g km\n',aa);

fprintf('orbital energy = %g kg km^2 s^-2\n',energy);

fprintf('h = %g m^2/s\n',h);

fprintf('p = %g km\n',p);

apoapsis altitude = 1869.59 km

orbital energy = -26.5806 kg km^2 s^-2

h = 54394.8 m^2/s

p = 7422.96 km

Problem 13:

Function:

function [nu, e, energy, p, h, rp, ra] = flightPathSpeedRadius(v,r,gamma,mu)

%This functions calculates true anomaly, eccentricity, and total energy given speed, flight path angle, and radius

%Function call: [nu e energy p h rp ra] = flightPathSpeedRadius(v,r,gamma,mu)

%

%Input: v, speed

%Input: r, radius

%Input: gamma, flight path angle

%Input: mu, gravitational parameter

%

%Output: nu, true anomaly

%Output: e, eccentricity

%Output: energy, total mechanical energy

%Output: p, semi-latus rectum

%Output: h, magnitude of the specific angular momentum

%Output: rp, periapsis radius

%Output: ra, apoapsis radius

a = ((((v/sqrt(mu))^2)-(2/r))^-1)\*-1;

h = r\*v\*cos(gamma);

p = (h^2)/mu;

e=sqrt(1-(p/a));

nu = acos((p/(r\*e))-(1/e));

energy = -(mu)/(2\*a);

rp = p/(1+e);

ra = p/(1-e);

end

Main:

clc;clear;

v = 0.8;

gamma = 0;

altitude = 4000;

earthRadius = 6378.145;

r = altitude+earthRadius;

mu = 398600;

[nu, e, energy, p, h, rp, ra] = flightPathSpeedRadius(v,r,gamma,mu);

fprintf('orbital energy = %g kg km^2 s^-2\n',energy);

fprintf('h = %g m^2/s\n',h);

fprintf('p = %g km\n',p);

fprintf('rp = %g km\n',rp);

fprintf('ra = %g km\n',ra);

orbital energy = -38.0876 kg km^2 s^-2

h = 8302.52 m^2/s

p = 172.935 km

rp = 87.1938 km

ra = 10378.1 km

Problem 14:

Function:

function [hVec, eVec, hDote, p, a, nu] = positionVelocity(rVec, vVec, mu)

%This function takes postion and velocity vectors as inputs and calculates orbital quantities

%Function call: [hVec, eVec, hDote, p, a, nu] = positionVelocity(rVec, vVec, mu)

%

%Input: rVec, postion vector

%Input: vVec, velocity vector

%Input: mu, gravitational parameter

%

%Output: hVec, specific angular momentum vector

%Output: eVec, eccentricity vector

%Output: hDote, dot product of specific angular momentum vector and eccentricity vector

%Output: p, semi-latus rectum

%Output: a, semi-major axis

%Output: nu, true anomaly

hVec = cross(rVec,vVec);

r = norm(rVec);

eVec = (cross(vVec,hVec)/mu)-(rVec/r);

hDote = dot(hVec,eVec);

h = norm(hVec);

p = h^2/mu;

e = norm(eVec);

a = abs(p/(1-e^2));

nu = acos((p/(r\*e))-(1/e));

end

Main:

clc;clear;

rVec = [-0.6 -1 0.75];

vVec = [0.8 -0.45 0.45];

mu = 1;

[hVec, eVec, hDote, p, a, nu] = positionVelocity(rVec, vVec, mu);

fprintf('The specific angular momentum vector is: ');

fprintf('%g, %g, %g\n',hVec(1),hVec(2),hVec(3));

fprintf('The eccentricity vector is: ');

fprintf('%g, %g, %g\n',eVec(1),eVec(2),eVec(3));

fprintf('h in I dotted with the eccentricity vector = %g\n',hDote);

fprintf('p = %g km\n',p);

fprintf('a = %g km\n',a);

fprintf('true anomaly = %g rad\n',nu);

The specific angular momentum vector is: -0.1125, 0.87, 1.07

The eccentricity vector is: -0.440269, -0.185407, 0.104461

h in I dotted with the eccentricity vector = 5.55112e-17

p = 1.91446 km

a = 2.51612 km

true anomaly = 0.678355 rad

Problem 15:

Function:

function [e, ap, vp] = radiusSpeedTrueAnomaly(r,v,nu,rb,mu)

%This function takes radius, speed, and true anomaly as inputs and calculates periapsis altitude, periapsis speed, and eccentricity

%Function call: [e, ap, vp] = radiusSpeedTrueAnomaly(r,v,nu,mu)

%

%Input: r, radius

%Input: v, speed

%Input: rb, radius of body

%Input: nu, true anomaly

%

%Output: e, eccentricity

%Output: ap, periapsis altitude

%Output: vp, periapsis speed

a = ((((v/sqrt(mu))^2)-(2/r))^-1)\*-1;

eccen = @(e) a\*e^2+r\*e\*cos(nu)-a+r;

e = fzero(eccen,1);

rp = a\*(1-e);

ap = rp-rb;

vp = sqrt(mu)\*sqrt((2/rp)-(1/a));

end

Main:

clc;clear;

r = 403000;

trueAnomaly = 151;

nu = deg2rad(trueAnomaly);

v = 2.25;

mu = 398600;

rb=6378.145;

[e, ap, vp] = radiusSpeedTrueAnomaly(r,v,nu,rb,mu);

fprintf('e = %g\n',e);

fprintf('altitude at periapsis = %g km\n',ap);

fprintf('speed at periapsis = %g km/s\n',vp);

e = 1.08131

altitude at periapsis = 4129.52 km

speed at periapsis = 8.88554 km/s

Problem 16:

(a). True

(b). False

(c). True

(d). False

(e). False