A picture containing fireworks, dark, water, flying

Description automatically generated

College of Engineering

School of Aeronautics and Astronautics

AAE 532

Orbital Mechanics

PS 4

Conical Orbits and Kepler’s Equation

*Author:*

Tomoki Koike

*Supervisor:*

K. C. Howell

October 2nd, 2020 Friday

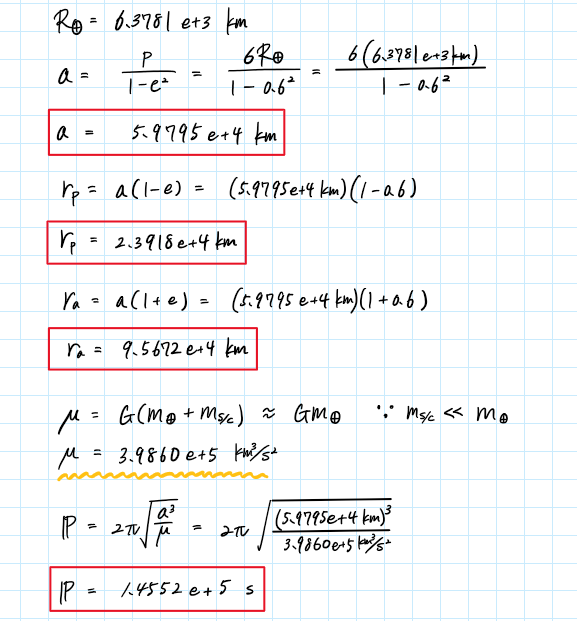
Purdue University

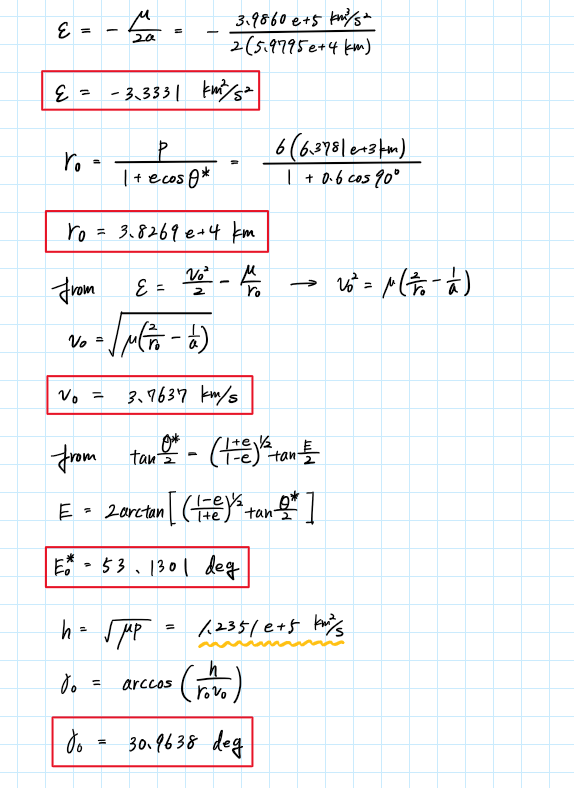
West Lafayette, Indiana

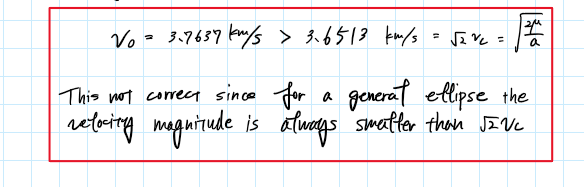
**Problem 1**: At some initial time, , a spacecraft is in orbit about the Earth. Its orbit is characterized by and . It is currently located at the point in the orbit such that .

1. Determine the following orbit parameters and spacecraft state information: . [Always list all angles in degrees.] Compare at this location with . Should ? Can your value be correct? How do you know?

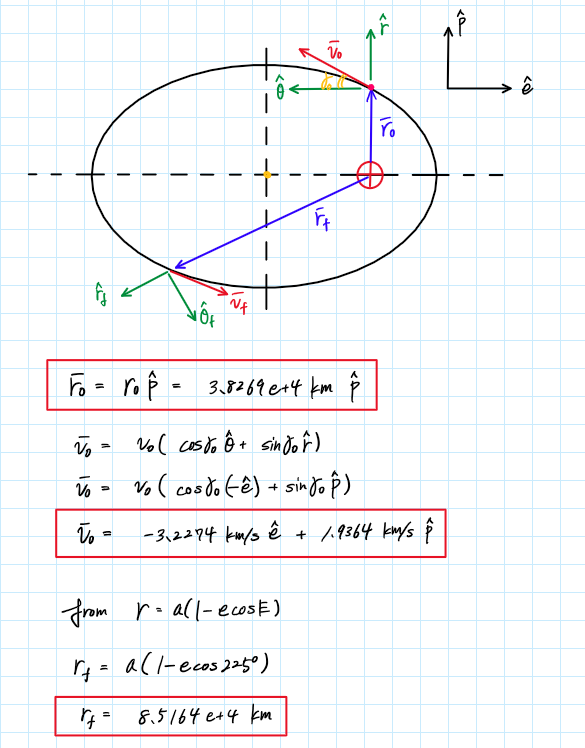
\*All calculations are handled using MATLAB (code is in appendix).

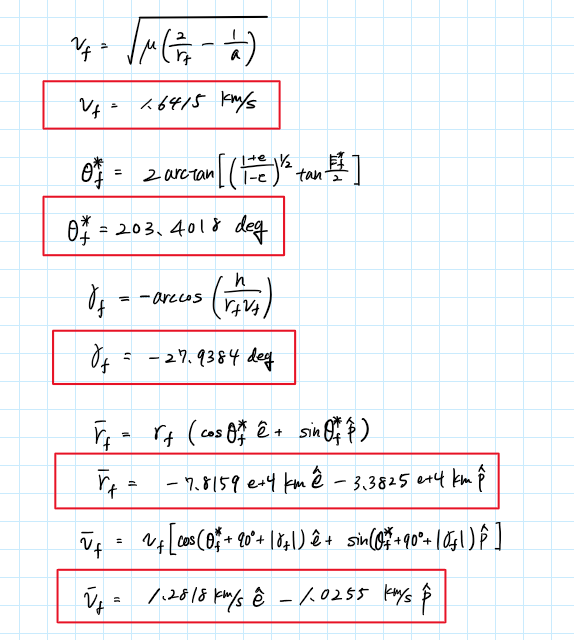




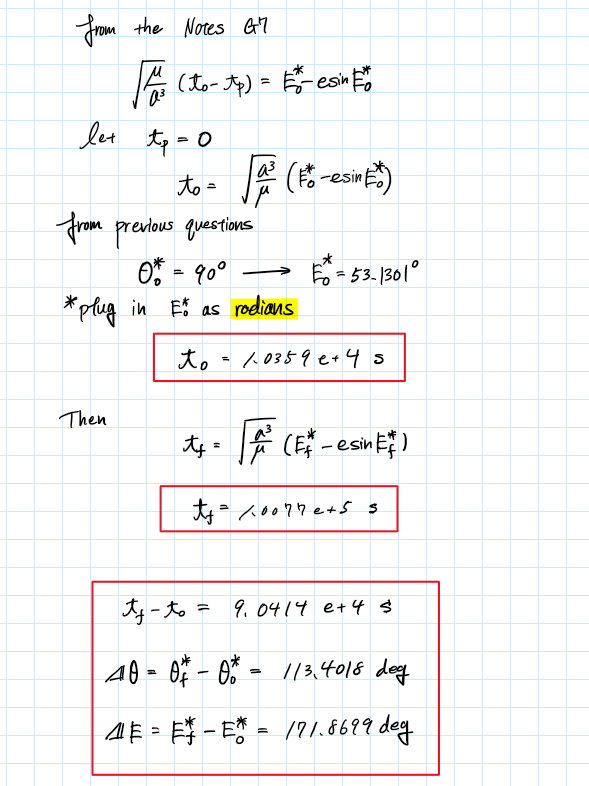


1. Write and in terms of components in the directions of and . At some later time, , determine when . Write and in terms of components in the directions of and .





1. Determine the time relative to periapsis, i.e., at ; also determine the final time relative to periapsis, i.e., when . What is the time-of-flight (TOF), i.e., as well as and ?



1. PLOT the entire orbit in MATLAB. (Do not use polar plots; compute and components along the path.) By hand, on the plot, mark the location of the satellite at and ; at each location, indicate ; also, sketch the local horizon. Indicate the arc used between and .

Chart

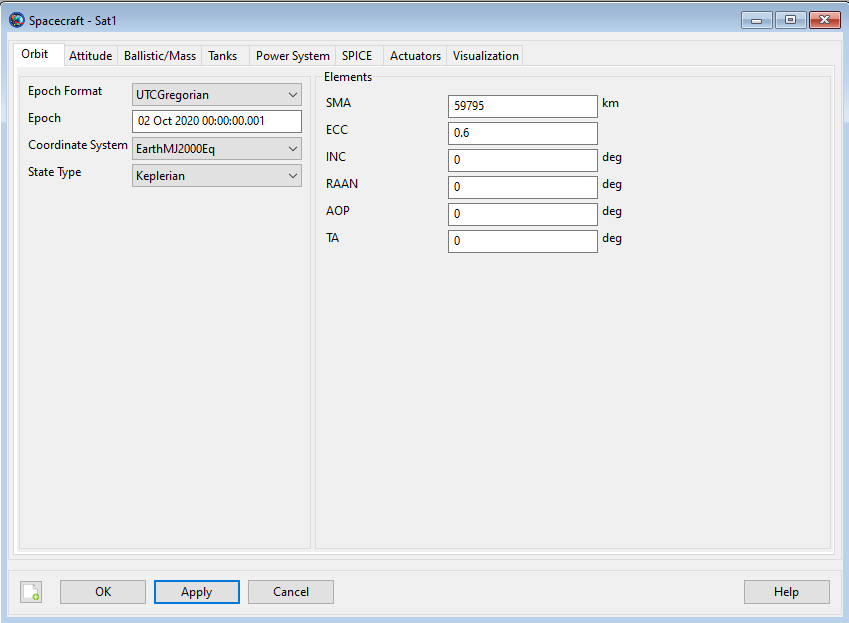
Description automatically generated

**Problem 2**: Return to Problem 1 and confirm your results in GMAT. Use October 2, 2020 as the start date.

1. What initial state can be input to GMAT for Sat1? Can you locate the rest of the quantities that were requested in Problem 1(a) and 1(b); do they confirm that your computations are correct? Can you determine the time difference ? Compare you MATLAB plot and the GMAT plot. Is your GMAT plot consistent with your MATLAB plot?

The initial conditions to Sat1 will be set as

* Epoch: UTCGregorian - 02 Oct 2020 00:00:00.001
* State Type: Keplerian
* SMA: 59795 km
* ECC: 0.6



The propagation for the 2 eccentric anomalies and are the following

A picture containing dark, light, lit, wire

Description automatically generated

A picture containing sitting, light, small, dark

Description automatically generated

From the report generated by GMAT we are able to confirm the quantities and when . We

For the orbit up to

|  |  |
| --- | --- |
| periapsis [km] | 23918 |
| apoapsis [km] | 95672 |
| energy [kJ] | -3.333058 |
| semi-major axis [km] | 59795 |
| semi-latus rectum [km] | 38269 |
| period | 145515 |
| [km] | 0.0020326 |
| [km] | 38269 |
| [km] | 0 |
| = [km] | 38269 |
| [km/s] | -3.2274 |
| [km/s] | 1.9364 |
| [km/s] | 0 |
| [km/s] | 3.7637 |
| [deg] | 90 |
| time-elapsed | 10359 |

For the orbit up to

|  |  |
| --- | --- |
| [km] | -78158 |
| [km] | -33825 |
| [km] | 0 |
| = [km] | 85163 |
| [km/s] | 1.2818 |
| [km/s] | -1.0255 |
| [km/s] | 0 |
| [km/s] | 1.6415 |
| [deg] | 203.40 |
| time-elapsed | 100773 |

The table above all agree with the results in Problem 1. Therefore, we can say that the GMAT result confirms our answers.

From the time elapse data,

This agrees with our calculations in Problem 1.

1. Also print out the data from GMAT. (You can submit output from the file generated in the propagate window under the Mission Sequence. Cut-and-paste the sections with the required data into a Word document. Highlight the requested quantities. You can also create a Report file; you may not want to include the entire file but, again cut-and-paste.)

The constants from the report



On the next, page there is a table for the velocity, position, and elapse-time. The specific time for and are highlighted.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sat1. VX** | **Sat1. VY** | **Sat1.VZ** | **Sat1.X** | **Sat1. Y** | **Sat1.Z** | **Sat1.ElapsedSecs** |
| 0 | 5.163764747 | 0 | 23918 | 0 | 0 | 0 |
| -0.04180402 | 5.163493991 | 0 | 23916.74585 | 309.8204696 | 0 | 59.99999988 |
| -0.162518739 | 5.159670198 | 0 | 23899.02547 | 1205.004061 | 0 | 233.4193771 |
| -0.512376379 | 5.122832596 | 0 | 23727.50067 | 3815.381452 | 0 | 740.8400658 |
| -0.897720259 | 5.036396539 | 0 | 23319.18205 | 6752.969377 | 0 | 1318.718584 |
| … | … | … | … | … | … | … |
| -3.163624424 | 2.574602722 | 0 | 6764.817645 | 33534.38749 | 0 | 8249.895158 |
| -3.209227423 | 2.277976586 | 0 | 3808.322783 | 35781.71315 | 0 | 9176.954888 |
| -3.226977383 | 1.985647356 | 0 | 578.5224334 | 37917.2734 | 0 | 10179.8525 |
| -3.227352967 | 1.936411951 | 0 | 0.002032598 | 38268.79878 | 0 | 10359.11481 |
| … | … | … | … | … | … | … |
| 0.625639348 | -1.229718757 | 0 | -91260.32255 | -18033.3843 | 0 | 86948.02516 |
| 0.748572524 | -1.202926741 | 0 | -89405.56726 | -21318.67931 | 0 | 89648.02516 |
| 0.873611317 | -1.17045327 | 0 | -87216.13658 | -24524.07901 | 0 | 92348.02516 |
| 1.001153939 | -1.131730649 | 0 | -84685.81353 | -27633.50716 | 0 | 95048.02516 |
| 1.131626053 | -1.086042462 | 0 | -81807.26968 | -30629.16103 | 0 | 97748.02516 |
| 1.265483661 | -1.032486715 | 0 | -78571.98686 | -33491.06266 | 0 | 100448.0252 |
| 1.281831493 | -1.025465192 | 0 | -78158.44993 | -33825.16003 | 0 | 100772.7109 |

1. Add an X-Y plot to the output. Plot speed as a function of elapsed time in seconds. Print the plot. Mark your time that you computed in Problem 1. Does the max velocity location in your plot correlate to the periapsis time in the GMAT plot?

Chart, line chart

Description automatically generated

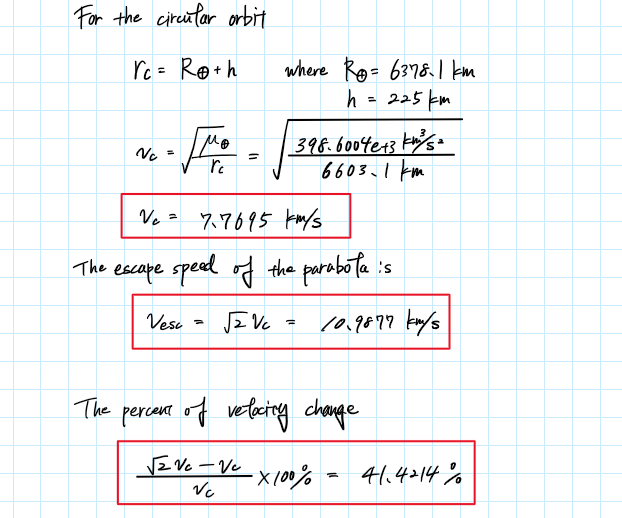
apoapsis

periapsis

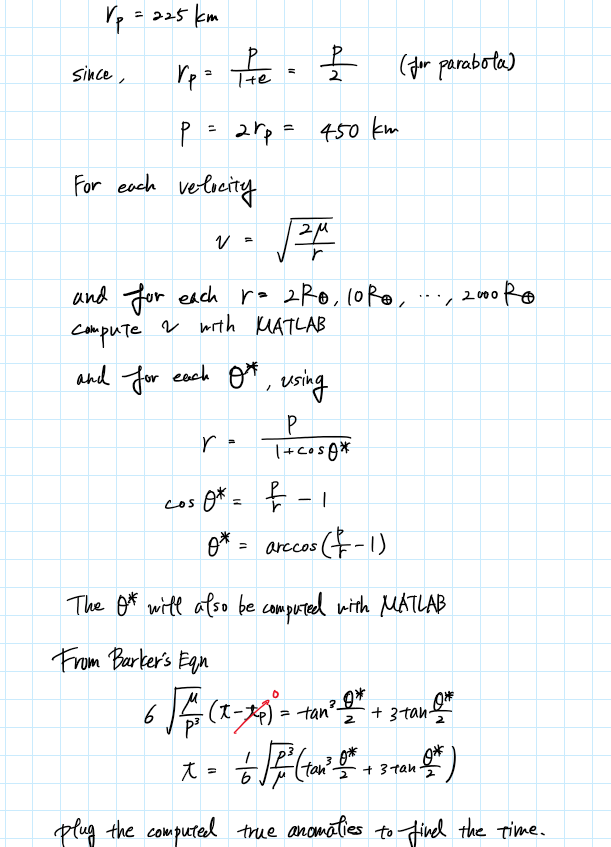
From this plot we can tell that the periapsis which is the closest to the Earth has the largest velocity magnitude. Thus, we can say that the GMAT plot agrees with the theoretical results from Kepler’s equations.

**Problem 3:** To investigate the requirements for departure, assume that a spacecraft is departing the vicinity of the Earth along a parabolic path. Consider the spacecraft to be located at perigee on the parabola.

1. A circular parking orbit about the Earth may be defined at 225 km altitude. At a perigee altitude of 225 km on the parabola, compare the escape velocity on the parabola with the relative velocity in a circular orbit with the same altitude. To shift from the circular orbit to the escape trajectory, what % increase in velocity is required.



1. Compute the velocity along the parabola as it departs the vicinity of the Earth, that is, at the following distances: one additional distance of your choice . Determine the true anomaly that corresponds to each distance. Also include the time since passing the periapsis at each distance (in days).



The results computed from MATLAB are tabulated below.

|  |  |  |  |
| --- | --- | --- | --- |
| Distances | Velocity [km/s] | True Anomaly [deg] | Time since [days] |
|  | 7.9054 | 164.7360 | 0.012776 |
|  | 3.5354 | 173.1899 | 0.13994 |
|  | 1.2909 | 177.5146 | 2.8612 |
|  | 0.79054 | 178.4781 | 12.4541 |
|  | 0.39527 | 179.2391 | 99.6129 |
|  | 0.24999 | 179.5187 | 393.7389 |

1. In the MATLAB script from the first problem, plot the parabola corresponding for altitude 225 km between . Mark on the plot, at ; also, sketch the l.h. (local horizon). Also add the directrix. Compare and is there a pattern?

Chart

Description automatically generated

Chart, line chart

Description automatically generated

From the graph above, we can see the relationship between the flight path angle and the true anomaly for a parabola is

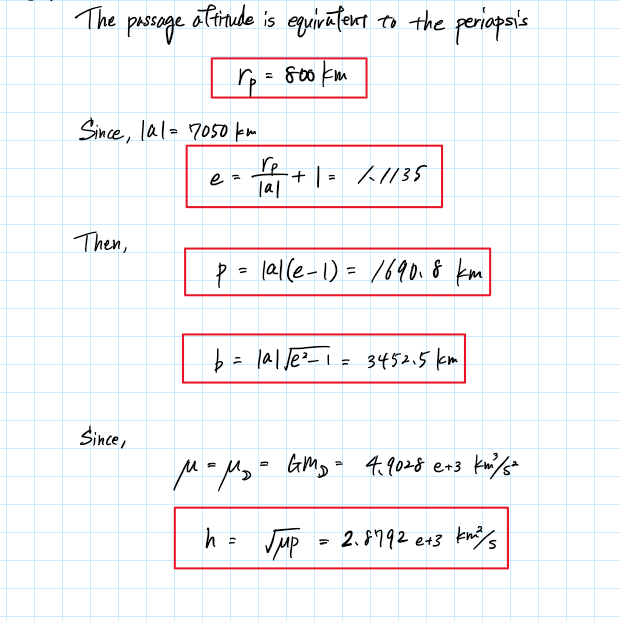
1. At , is it reasonable to model the problem as a two-body problem (Earth and spacecraft)?

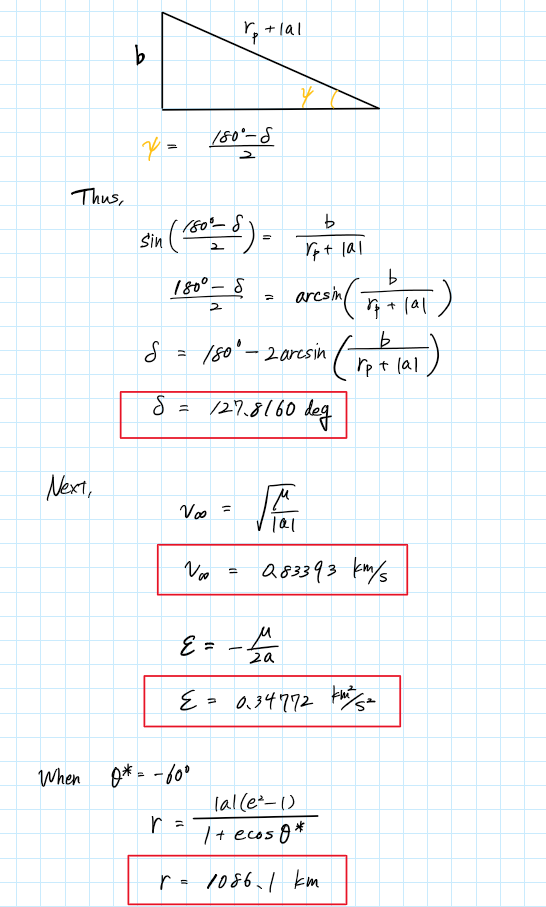
Since the distance of the Moon from the Earth is , we can say that we cannot model this as a two-body problem. The spacecraft is farther away from than the Moon and is possible for the spacecraft to be near the Moon. This puts the spacecraft under the influence of the Moon’s gravitation, and therefore, we must model the problem as a three-body problem rather than a two-body problem.

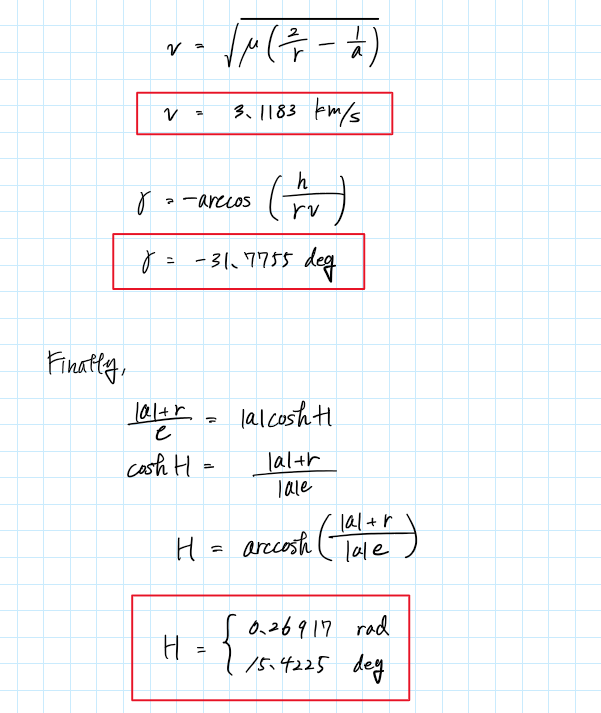
**Problem 4**: As part of the new lunar initiative, an unmanned probe is approaching the Moon on a hyperbola. The hyperbola is defined such that and the passage altitude is 800 km altitude. At the “current” time, the probe is located at .

1. Determine the following additional orbital characteristics: **.** Determine the following quantities at the current time: , time till perilune.

\*All calculations are done in MATLAB (code in appendix).







1. Use your MATLAB script and plot the hyperbola between . Mark the probe at and label , aim point, . (Always include the local horizon!)

Since the hyperbola equation is defined as

The asymptotes are expressed as

Keeping this in mind we plot the following graph

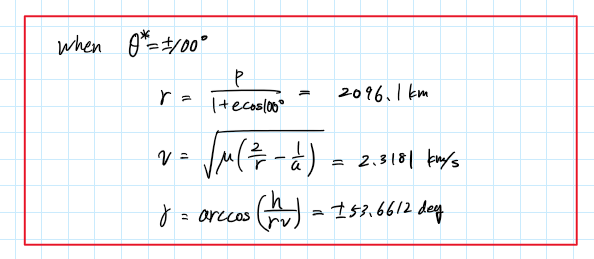
Chart, line chart

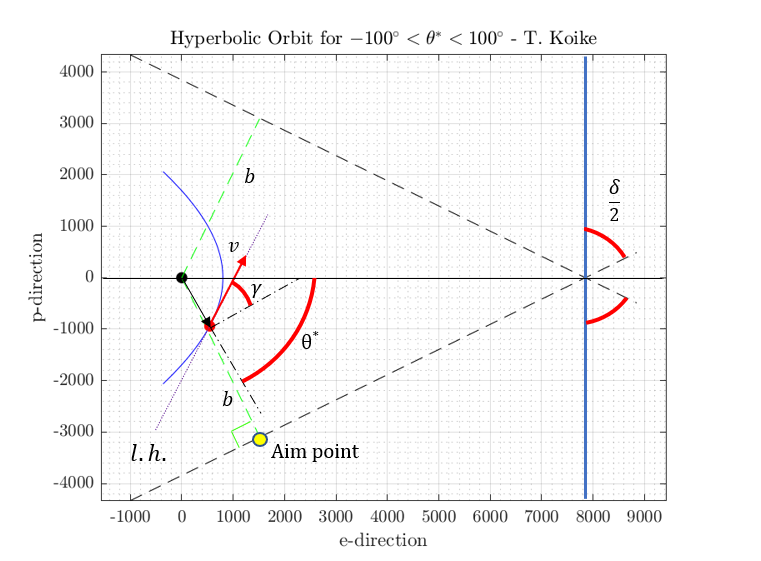
Description automatically generated

Aim point

The red dot indicated in the plot represents the point at .

1. Determine at ; add this information to the plot.





Appendix

MATLAB Code

Problem 1

%% AAE 532 HW 4 Problem 1

% Tomoki Koike

close all; clear all; clc;

fdir = 'C:\Users\Tomo\Desktop\studies\2020-Fall\AAE532\matlab\outputs\ps4';

set(groot, 'defaulttextinterpreter','latex');

set(groot, 'defaultAxesTickLabelInterpreter','latex');

set(groot, 'defaultLegendInterpreter','latex');

format shortEng;

% Set constants

planet\_consts = setup\_planetary\_constants(); % Function that sets up all the constants in the table

earth = planet\_consts.earth; % structure of earth

G = 6.6743015e-20; % Gravitational constant [km^3/kg/s^2]

% (a)

r\_E = earth.mer; % radius of earth [km]

e = 0.6; % eccentricity

theta\_star = 90; % true anomaly [deg]

p = 6\*r\_E; % latus rectum [km]

a = p/(1 - e^2); % semi mahor axis [km]

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

rp = a\*(1 - e); % periapsis [km]

ra = a\*(1 + e); % apoapsis [km]

mu = earth.gp; % gravitational parameter [km^3/s^2]

IP = 2\*pi\*sqrt(a^3 / mu); % period [s]

En = -mu / 2 / a; % specific energy [km^2/s^2]

r0 = p;

v0 = sqrt(mu \* (2/r0 - 1/a));

E0 = trueAnomaly2EccAnomaly(e, theta\_star, "deg"); % eccentric anomaly [deg]

h = sqrt(mu \* p); % specific angular momentum [km^2/s]

gamma0 = acosd(h / r0 / v0); % flight path angle [deg]

vc = sqrt(mu / a )

% (b)

r0\_vec = r0 \* [0, 1, 0];

v0\_vec = v0 \* [-cosd(gamma0), sind(gamma0), 0];

Ef = 225; % eccentric anomaly at a certain time tf [deg]

rf = a \* (1 - e\*cosd(Ef));

vf = vis\_viva(rf, a, mu);

theta\_star\_f = eccAnomaly2trueAnomaly(e, Ef, "deg");

gamma\_f = acosd(h / rf / vf);

rf\_hat = [cosd(theta\_star\_f), sind(theta\_star\_f), 0];

vf\_hat = [cosd(theta\_star\_f+90+gamma\_f), sind(theta\_star\_f+90+gamma\_f), 0];

rf\_vec = rf \* rf\_hat;

vf\_vec = vf \* vf\_hat;

% (c)

t0 = ellipse\_time\_KeplerEqn(0, E0, "deg", mu, a, e);

tf = ellipse\_time\_KeplerEqn(0, Ef, "deg", mu, a, e);

Dt = tf - t0;

Dtheta = theta\_star\_f - theta\_star;

DE = Ef - E0;

% (d)

% Arrow drawing function

drawArrow = @(x,y,varargin) quiver( x(1),y(1),x(2)-x(1),y(2)-y(1),0, varargin{:} );

% Plotting

fig1 = figure("Renderer","painters");

hold on; grid on; grid minor; box on; axis equal;

% ylim([-5.5e+4, 5.5e+4]); xlim([-7e+4, 7e+4])

tt = 0:0.01:2\*pi;

X = a \* cos(tt); Y = b \* sin(tt);

% Ellipse

plot(X, Y, '-k')

% Axes

x\_axis = linspace(-a, a, 2^9);

y\_axis = linspace(-b, b, 2^9);

plot(x\_axis, zeros(size(x\_axis)), '--k')

plot(zeros(size(y\_axis)), y\_axis, '--k')

% Center

plot(a\*e, 0, '.r', 'MarkerSize', 25)

% t0 point

rx\_t0 = [a\*e, a \* cosd(E0)]; ry\_t0 = [0, b \* sind(E0)];

drawArrow(rx\_t0, ry\_t0, 'linewidth',2,'Color',[0 0 1]);

rx\_t0E = [0, a \* cosd(E0)]; ry\_t0E = [0, b \* sind(E0)];

drawArrow(rx\_t0E, ry\_t0E, 'linewidth',1,'Color',[0 1 0]);

plot(a \* cosd(E0), b \* sind(E0), '.b', 'MarkerSize', 20)

% tf point

rx\_tf = [a\*e, a \* cosd(Ef)]; ry\_tf = [0, b \* sind(Ef)];

drawArrow(rx\_tf, ry\_tf, 'linewidth',2,'Color',[0 0 1]);

rx\_tfE = [0, a \* cosd(Ef)]; ry\_tfE = [0, b \* sind(Ef)];

drawArrow(rx\_tfE, ry\_tfE, 'linewidth',1,'Color',[0 1 0]);

plot(a \* cosd(Ef), b \* sind(Ef), '.b', 'MarkerSize', 20)

% Origin

plot(0, 0, '.k', 'MarkerSize', 18)

hold off

title('Elliptical Orbit Around Earth with $e$=0.6 - T. Koike')

saveas(fig1, fullfile(fdir, 'p1\_orbitplot.png'));

Problem 3

%% AAE 532 HW 4 Problem 3

% Tomoki Koike

close all; clear all; clc;

fdir = 'C:\Users\Tomo\Desktop\studies\2020-Fall\AAE532\matlab\outputs\ps4';

set(groot, 'defaulttextinterpreter','latex');

set(groot, 'defaultAxesTickLabelInterpreter','latex');

set(groot, 'defaultLegendInterpreter','latex');

format shortEng;

% Set constants

planet\_consts = setup\_planetary\_constants(); % Function that sets up all the constants in the table

earth = planet\_consts.earth; % structure of earth

G = 6.6743015e-20; % Gravitational constant [km^3/kg/s^2]

% (a)

h = 225; % altitude [km]

r\_E = earth.mer; % radius of earth [km]

mu = earth.gp;

r\_c = r\_E + h;

v\_c = sqrt(mu / r\_c);

v\_esc = sqrt(2) \* v\_c;

% (b)

r\_p = h;

p = 2 \* r\_p

dist = r\_E \* [2, 10, 75, 200, 800, 2000];

v\_para = []; TA = []; t\_rp = [];

for ri = dist

v\_para = [v\_para, parabola\_for\_r(ri, mu)];

temp = true\_anomaly\_calc(p, ri);

TA = [TA, temp];

t\_rp = [t\_rp, backer\_eqn\_time(p, mu, temp)];

end

t\_rp = t\_rp / 60 / 60/ 24;

% (c)

% Plotting

TA = -140:0.01:140;

R = p ./ (1 + cosd(TA));

X = R .\* cosd(TA);

Y = R .\* sind(TA);

% Find point for -120 degrees

R\_n120= R(TA == -120);

fig1 = figure("Renderer","painters");

title('Parabolic Orbit for $-140^{\circ}<\theta^\*<140^{\circ}$ - T. Koike')

xlabel('x-direction')

ylabel('y-direction')

plot(X, Y, '-b')

grid on; grid minor; box on; axis equal;

hold on;

plot(R\_n120\*cosd(-120), R\_n120\*sind(-120), '.r', 'MarkerSize', 20)

% Directrix

direx\_y = fig1.CurrentAxes.YLim(1):fig1.CurrentAxes.YLim(2);

direx\_x = ones(size(direx\_y))\*2\*r\_p;

plot(direx\_x, direx\_y, '-k')

% x-axis

xaxis\_x = fig1.CurrentAxes.XLim(1):fig1.CurrentAxes.XLim(2);

xaxis\_y = zeros(size(xaxis\_x));

plot(xaxis\_x, xaxis\_y, '--k')

% Center

plot(0, 0, '.k', 'MarkerSize', 20)

hold off

saveas(fig1, fullfile(fdir, 'p3\_plot.png'))

% Plotting gamma with TA

% gamma

rs = calc\_parabola\_r(p, TA);

vs = parabola\_for\_r(rs, mu);

gammas = acosd(sqrt(mu\*p) ./ rs ./ vs);

idx1 = find(TA == -140);

idx2 = find(TA==0);

gammas(idx1:idx2) = -1\*gammas(idx1:idx2);

fig2 = figure("Renderer","painters");

plot(TA, gammas)

grid on; grid minor; box on; axis equal;

title('Flight Path Angle Over True Anomaly - T. Koike')

xlabel('$\theta^\*$ [deg]')

ylabel('$\gamma$ [deg]')

saveas(fig2, fullfile(fdir, 'p3\_gamma\_theta.png'))

function v = parabola\_for\_r(r,mu)

v = sqrt(2 \* mu ./ r);

end

function r = calc\_parabola\_r(p, TA)

r = p ./ (1 + cosd(TA));

end

function TA = true\_anomaly\_calc(p, r)

TA = acosd(p/r - 1);

end

function et = backer\_eqn\_time(p, mu, TA)

et = sqrt(p^3 / mu) / 6 \* (tand(TA/2)^3 + 3\*tand(TA/2));

end

Problem 4

%% AAE 532 HW 4 Problem 4

% Tomoki Koike

close all; clear all; clc;

fdir = 'C:\Users\Tomo\Desktop\studies\2020-Fall\AAE532\matlab\outputs\ps4';

set(groot, 'defaulttextinterpreter','latex');

set(groot, 'defaultAxesTickLabelInterpreter','latex');

set(groot, 'defaultLegendInterpreter','latex');

format shortEng;

% Set constants

planet\_consts = setup\_planetary\_constants(); % Function that sets up all the constants in the table

moon = planet\_consts.moon; % structure of moon

G = 6.6743015e-20; % Gravitational constant [km^3/kg/s^2]

% Find characterisitics

mu = moon.gp; % gravitational parameter

a = 7050; % semi major axis [km]

r\_p = 800;

e = r\_p/a + 1;

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

b = a \* sqrt(e^2 - 1)

h = sqrt(mu \* p)

FBA = 180 - 2\*asind(b / (r\_p + a))

v\_inf = sqrt(mu / a)

En = mu / 2 / a

% at TA = -60 deg current time

TA\_curr = -60;

r\_curr = p / (1 + e\*cosd(TA\_curr));

v\_curr = vis\_viva(r\_curr, -a, mu);

FPA\_curr = -acosd(h / r\_curr / v\_curr);

H\_curr = acosh((a + r\_curr) / a / e)

% (b) Plotting

% Hyperbola

TA = -100:1:100;

R = p ./ (1 + e\*cosd(TA));

X = R .\* cosd(TA);

Y = R .\* sind(TA);

fig1 = figure("Renderer","painters");

plot(X, Y, '-b')

title('Hyperbolic Orbit for $-100^{\circ}<\theta^\*<100^{\circ}$ - T. Koike')

xlabel('e-direction')

ylabel('p-direction')

grid on; grid minor; box on; axis equal; hold on;

% Asymptotes

x\_asymp = -1000:(a\*e+1000);

y\_asymp\_p = sqrt(e^2 - 1)\*(x\_asymp - a\*e);

y\_asymp\_n = -sqrt(e^2 - 1)\*(x\_asymp - a\*e);

plot(x\_asymp, y\_asymp\_p, '--k')

plot(x\_asymp, y\_asymp\_n, '--k')

% Center

plot(0, 0, '.k', 'MarkerSize', 20)

% b line

x\_b = 0:1513;

y\_bp = 1/sqrt(e^2-1) .\* x\_b;

y\_bn = -1/sqrt(e^2-1) .\* x\_b;

plot(x\_b, y\_bp, '--g')

plot(x\_b, y\_bn, '--g')

% Plot TA=-60

Xi = X(TA==-60);

Yi = Y(TA==-60);

plot(Xi, Yi, '.r', 'MarkerSize', 20)

hold off

saveas(fig1, fullfile(fdir, 'p4\_hyperbola.png'))

% at TA = 100 deg current time

TA\_curr = 100;

r\_curr = p / (1 + e\*cosd(TA\_curr));

v\_curr = vis\_viva(r\_curr, -a, mu);

FPA\_curr = acosd(h / r\_curr / v\_curr);