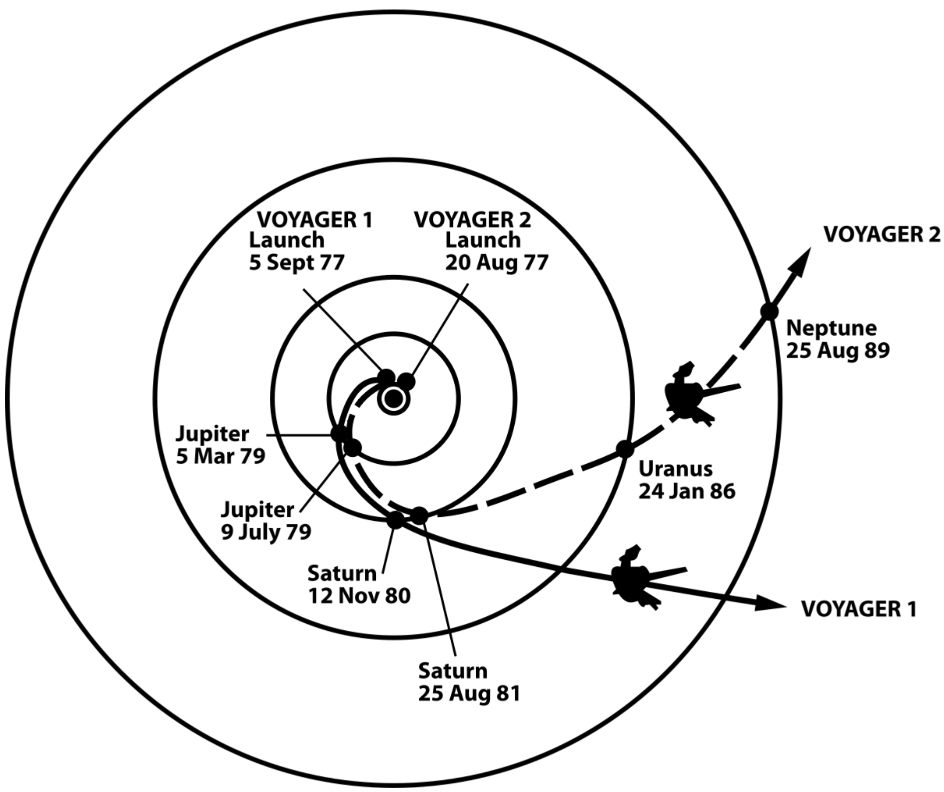
Voyager 2 Mission Simulation

Aerospace 343: Orbital Mechanics
Undergraduate Aerospace Engineering
University of Michigan - Ann Arbor

By: Dan Card, dcard@umich.edu
Date: December 13, 2019

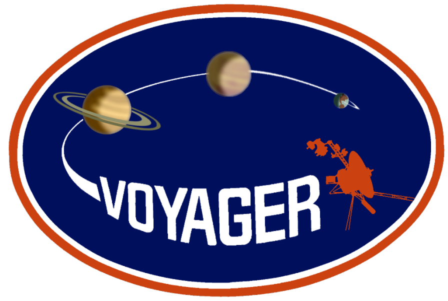


Contents

1 Introduction

Voyager 2 was launched on August 20, 1977, with a primary focus of conducting multiple flybys across the Solar System. Voyager 2 is currently the only spacecraft to have visited both Uranus and Neptune. Voyager 2 flew by Jupiter, Saturn, Uranus, and Neptune.

During Voyager 2's main mission it was able to visit all four gas giant planets and photographed many of the planets' moons. After conducting the main mission of visiting the gas giants it made its way to interstellar space on November 5, 1998. After entering interstellar space, engineers turned off all nonessential instruments to conserve power. It is estimated that Voyager 2's instruments will remain operational until the year 2025.[1]



Mission Badge: Voyager 2

Scientific Instrument(s)

- Imaging system
- Infrared interferometer spectrometer
- Ultraviolet spectrometer
- Triaxial fluxgate magnetometer
- Plasma spectrometer
- Low-energy charged particles detectors
- Cosmic Ray System (CRS)
- Photopolarimeter System (PPS)
- Plasma Wave System (PWS)

Type: Flyby
Status: Current
Launch Date: August 20, 1977 10:29 a.m. EDT(14:29 UTC)
Launch Location: Cape Canaveral Air Force Station Florida
Target: Jupiter, Saturn, Uranus, Neptune
Destination: Jupiter, Saturn, Uranus, Neptune

Simulating Voyager 2 Mission

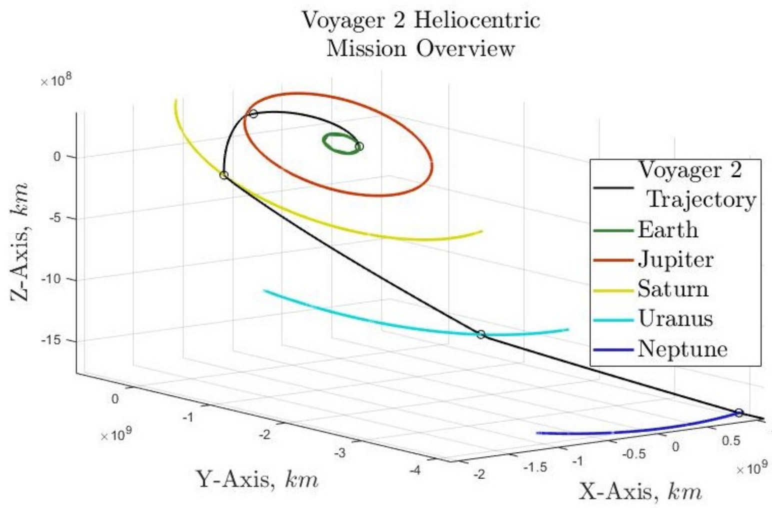
The goal of this project is to simulate Voyager 2's mission through the solar system stepping through all the numerous flybys of the four gas giants. This project will simulate the mission by using Lamberts Problem* that can be found in Appendix B to solve for the velocity needed to calculate the corrective burns throughout the mission to account for perturbations and altered trajectories after gravity-assist. After using the Lamberts method to determine the velocity needed to conduct a given burn the simulation will forward propagate and numerically determine Voyager 2's trajectory as it transits across the Solar System. This project analyzes the trajectories from Voyager 2, the velocities and in-depth analysis into the possibilities that arise from simplifications and or numerical inaccuracies.

*A numerical method to find the orbital elements given two initial points.

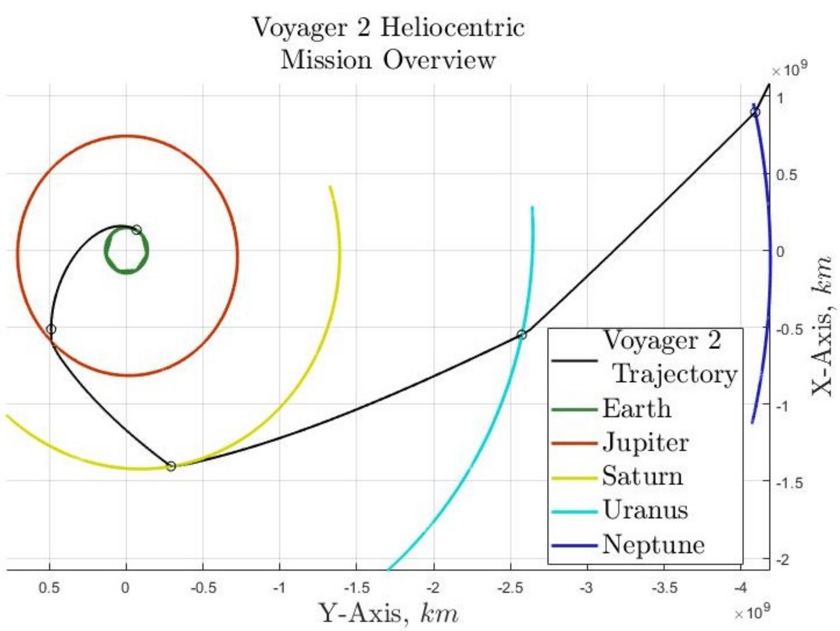
2 Simulation Results

A Voyager 2's Trajectory Across The Solar System

After applying the Lamberts method to determine the spacecraft's velocity after each interplanetary transfer, my simulation would forward propagate the spacecraft until it reached a new target body that it would perform a gravity assist on. The simulation and all the flybys are shown below in Figure 1a,1b. These plots show Voyager 2's trajectory resolved into a Heliocentric frame to show its path relative to the orbits of the planets.



(a) Voyager 2 trajectory across the solar system.

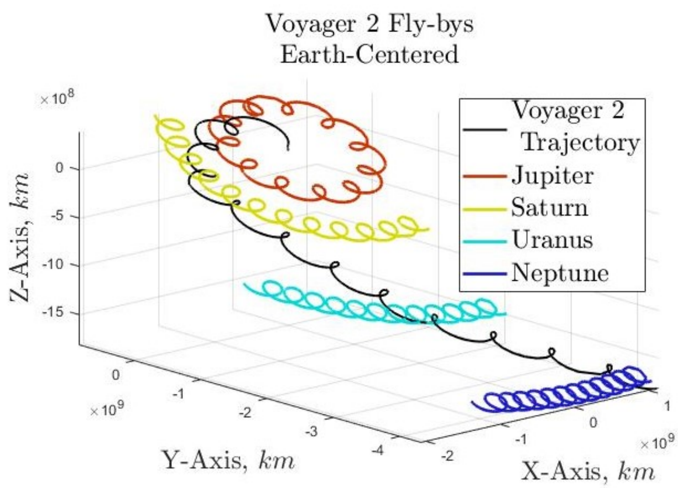


(b) Top-Down view of Voyager 2's trajectory across the solar system.

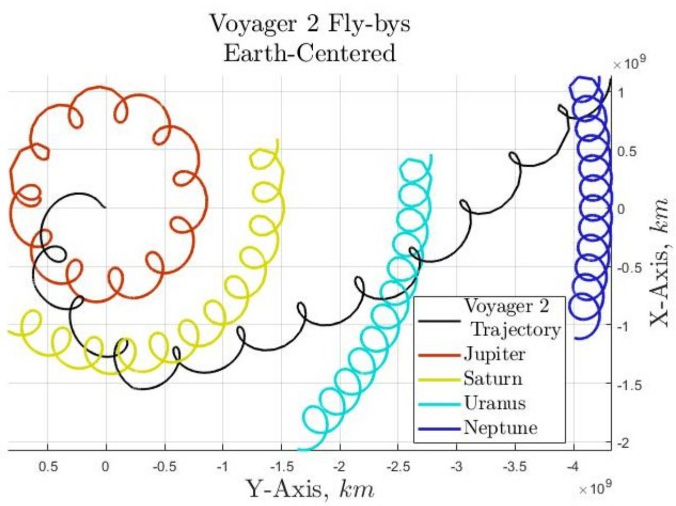
Figure 1: Voyager 2 Deep Space Mission flight trajectory, from a heliocentric frame.

B Voyager 2's Trajectory From Earth

Below in Figure 2a, 2b is how Voyager 2's trajectory appears relative to Earth in an Earth-Centered Earth Fixed Frame(ECEF). These trajectories appear to be oscillating but this results from adjusting the positions and the velocities to be relative to Earth. Resolving Figures 1a, 1b into Earth's frame



(a) Voyager 2 trajectory across the solar system from Earth's frame.



(b) Top-Down view Voyager 2 trajectory across the solar system from Earth's frame.

Figure 2: Voyager 2 Deep Space Mission flight trajectory, from a heliocentric frame.

C Voyager 2's Velocity Profile Across The Solar System

Voyager 2's main mission was to flyby the four giant gas planets; Jupiter, Saturn, Uranus, and Neptune then continue to interstellar space. Through each flyby, Voyager two was able to get a "boost" in velocity by conducting a gravity assist to continue its mission from one gas giant to another. Below in Figure 3, is the velocity profile of Voyager 2 as it transited across the Solar System with each spike indicating a gravity assist from each gas giant.

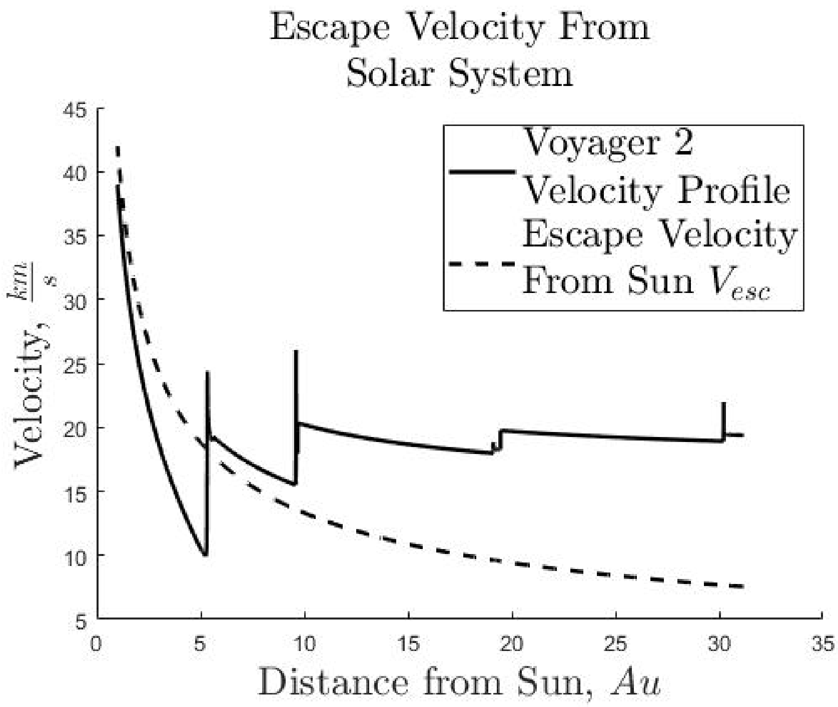
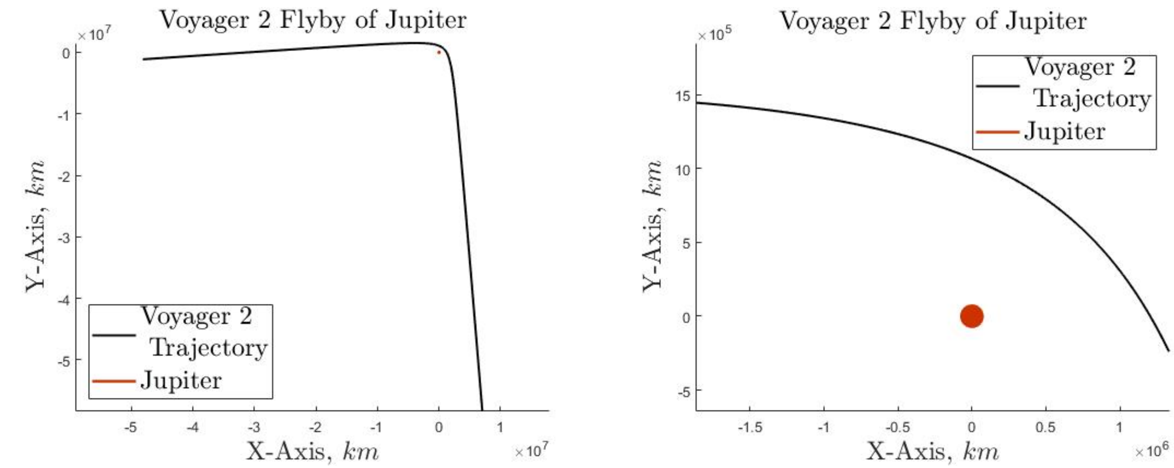


Figure 3: Voyager 2 velocity profile and Sun's escape velocity versus distance.

D Gravity Assists Across the Solar System

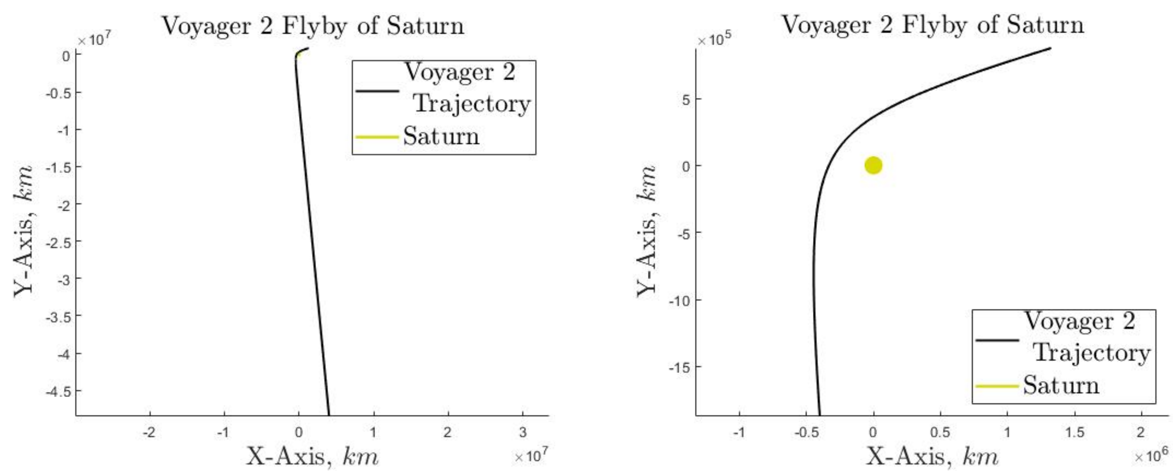
The gravity assists act to redirect Voyager 2 from one gas giant to the next until it ultimately leaves the Solar System. How a gravity assist works are when a satellite comes into proximity of a planet it can use its gravitational pull to increase its speed until it leaves the sphere of influence of the planet. These gravity assists can be shown below in Figures 4a, 5a, 6a, 7a.



(a) Jupiter flyby, entering and exiting sphere of influence.

(b) Close view of Jupiter's periapsis of flyby.

Figure 4: Voyager 2's flybys of Jupiter.

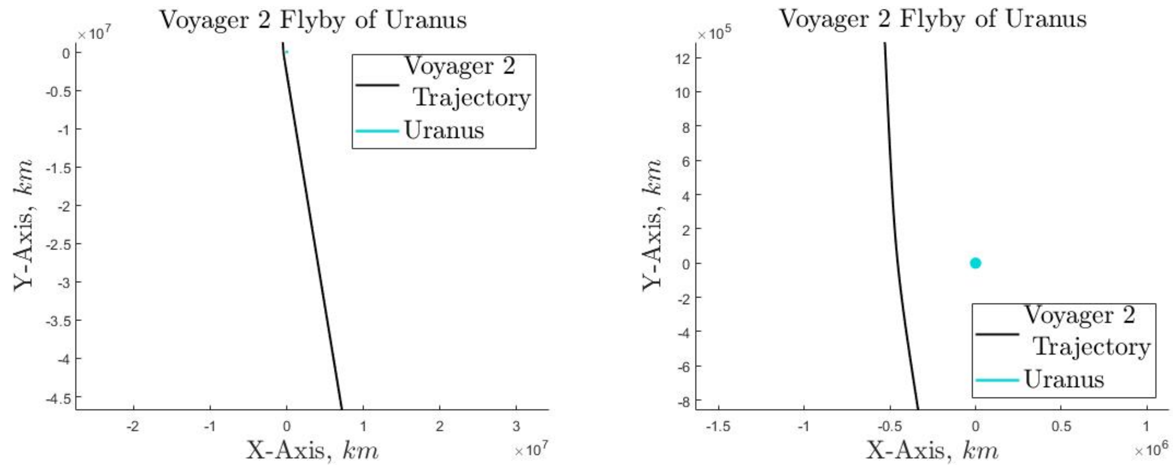


(a) Saturn flyby, entering and exiting sphere of influence.

(b) Close view of Saturn's periapsis of flyby.

Figure 5: Voyager 2's flybys of Saturn.

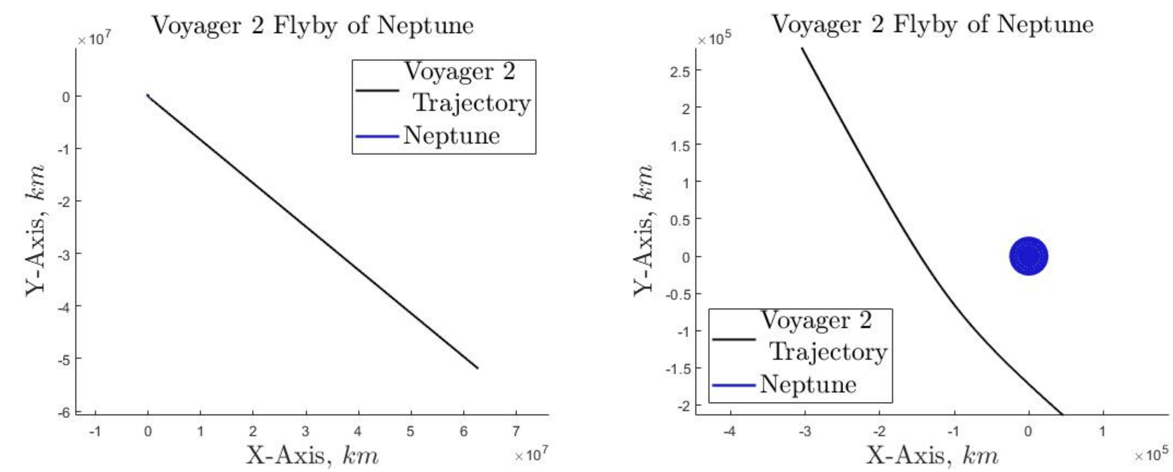
Gravity Assists Across the Solar System (Continued)



(a) Uranus flyby, entering and exiting sphere of influence.

(b) Close view of Uranus's periapsis of flyby.

Figure 6: Voyager 2's flybys of Uranus.



(a) Neptune flyby, enter and exit sphere of influence.

(b) Close view of Neptune's periapsis of flyby.

Figure 7: Voyager 2's flybys of Neptune.

3 Mission Analysis

A Analysis of ΔV Gained From Gravity Assists

When the satellite performs its flybys of the gas giants it will gain a velocity "boost" which is known as a gravity assist. Defining the gravity assist can be done using Equation 1

$$\Delta V = |\vec{v}_{\infty,2}| - |\vec{v}_{\infty,2}| \tag{1}$$

Where $|\vec{v}_{\infty,1/2}|$ is the magnitude of the velocity of the satellite once it has left the sphere of influence of the planet. This difference of ΔV is generated for every hyperbolic flyby for each planet along with the transfer and is shown in Table 1 below.

Table 1: ΔV imparted from gravity-assist.

Target Planet	ΔV, $\frac{km}{s}$
Jupiter Fly-By	8.262
Saturn Fly-By	2.478
Uranus Fly-By	0.281
Neptune Fly-By	0.493
Total ΔV	11.513

B Correcting Voyager 2's Trajectory

Throughout Voyager 2's simulation, I had to perform a series of burns to realign the satellite such that it would be able to perform an interplanetary transfer from one planet to another. These burns are the result of the planet flybys skewing the trajectory of the satellite such that it would not be able to transit to another planet. The burns have shown in Table 2 is the amount of ΔV needed to re-align the satellite back onto its respective orbit.

Table 2: Corrective burns to re-align Voyager 2.

Corrective Burn After Target Planet Fly-By	ΔV, $\frac{km}{s}$
Earth Correction	0.095
Jupiter Correction	0.361
Saturn Correction	2.362
Uranus Correction	1.487
Total ΔV	4.305

For reference, NASA's DAWN Deep-Space Mission generated a total of approximately 11$\frac{km}{s}$ throughout its mission entirety.

C Voyager 2's Escape From Solar System

Looking to Figure 3, Voyager 2 reached escape velocity after its flyby of Jupiter. The escape velocity is defined in Equation 2, where the escape velocity scales with the inverse of the distance. In the figure, the satellite passes the required escape speed after its hyperbolic flyby of Jupiter.

$$v_{esc} = \sqrt{\frac{2\mu}{r}} \tag{2}$$

Appendices

A MATLAB Simulation Code

```
        %--------------------------------------------------------------------------
%       Dan Card,   Aerospace 343 - Simulating Voyager 2 Deep Space Mission
%--------------------------------------------------------------------------
clear all; clc; close all;
%                JULIAN DATE            X-POSITION[km]      Y-POSITION[km]    ...
    Z-POSITION[km]      VX [km/s]          VY [km/s]              VZ [km/s]
states = [  2443378.81193287,   132384620.9297295, -67520631.79283643, -28460623.59955546, ...
    16.60619039863231,   30.90409880024641,  16.94896022879032;
            2443980.98293758,  -513307082.200194,   487523732.919045,   227786347.591631, ...
                -10.6277066387729,    0.146979252537972, -0.499225978899256;
            2444842.47565972,  -1403761582.09172,  -291275970.488251,  -59922358.3390430, ...
                -15.1774624119501,   -17.5915428395588,  -6.76604497240708;
            2446455.04151620,  -549383830.242340,  -2573248593.64353,  -1119252972.75572,  ...
                7.45158431499535,    -15.9209373455824,  -7.41680295528321;
            2447763.49763889,   897252737.328015,  -4091587528.38236,  -1697019059.72079,  ...
                13.3381581699716,   -14.1263323637747,  -5.45818819644464];
%% Earth to Jupiter Segment
opts0.cBody = 'Sun'; opts0.tBody = 'Jupiter'; opts0.stopCond = 0;
t0 = states(1,1); r0 = states(1, 2:4); v0 = states(1, 5:end); Xsc_0 = [r0';v0'];
tf = states(2,1); rf = states(2, 2:4);

[v0,delv1] = calcburn(Xsc_0, rf, tf-t0, opts0); Xsc_0 = [Xsc_0(1:3); v0];
[T,X] = SCPROP(0, (tf-t0)*86400, Xsc_0, opts0); Tp = T(end); T = SEC2JULIAN(T);

Xsc_tot = X; tlin = T'; delv = delv1;
%% Jupiter Fly-By
clear opts0 fig
opts0.cBody = 'Jupiter'; opts0.stopCond = 1;
t0 = Tp; Xsc_0 = SUN2PLANET(Tp, X(end,:),'Jupiter');
tf = t0 + 1200*86400;

[T,X] = SCPROP(t0,tf, Xsc_0, opts0); Tp = T; T = SEC2JULIAN(T);
jupfly = X; X = PLANET2SUN(Tp, X, 'Jupiter'); Tp = T(end);

Xsc_tot = [Xsc_tot;X]; tlin = [tlin, T'];
grav1 = norm(X(end,4:6))- norm(X(1,4:6)); gravtot = grav1;
%% Jupiter to Saturn Segment
opts0.cBody = 'Sun'; opts0.tBody = 'Saturn'; opts0.stopCond = 0;
t0 = Tp; Xsc_0 = X(end,:)';
tf = states(3,1); rf = states(3, 2:4);

[v0,delv2] = calcburn(Xsc_0, rf, tf-t0, opts0); Xsc_0 = [Xsc_0(1:3); v0];
[T,X] = SCPROP(t0*86400, tf*86400, Xsc_0, opts0); Tp = T(end); T = ...
    SEC2JULIAN(T-states(1,1)*86400);

Xsc_tot = [Xsc_tot;X]; tlin = [tlin, T']; delv = delv + delv2;
%% Saturn Fly-By
clear opts0 fig
opts0.cBody = 'Saturn'; opts0.stopCond = 1;
t0 = Tp; Xsc_0 = SUN2PLANET(Tp-states(1,1)*86400, X(end,:),'Saturn');
tf = t0 + 1200*86400;

[T,X] = SCPROP(t0,tf, Xsc_0, opts0); Tp = T; T = SEC2JULIAN(T-states(1,1)*86400);
satfly = X; X = PLANET2SUN(Tp-states(1,1)*86400, X, 'Saturn'); Tp = T(end);

Xsc_tot = [Xsc_tot;X]; tlin = [tlin, T'];
grav2 = norm(X(end,4:6))- norm(X(1,4:6)); gravtot = grav2 + gravtot;
%% Saturn to Uranus Segment
opts0.cBody = 'Sun'; opts0.tBody = 'Uranus'; opts0.stopCond = 0;
t0 = Tp; Xsc_0 = X(end,:)';
```

```
tf = states(4,1); rf = states(4, 2:4);

[v0,delv3] = calcburn(Xsc_0, rf, tf-t0, opts0); Xsc_0 = [Xsc_0(1:3); v0];
[T,X] = SCPROP(t0*86400, tf*86400, Xsc_0, opts0); Tp = T(end); T = ...
    SEC2JULIAN(T-states(1,1)*86400);

Xsc_tot = [Xsc_tot;X]; tlin = [tlin, T']; delv = delv + delv3;
%% Uranus Fly-By
clear opts0 fig
opts0.cBody = 'Uranus'; opts0.stopCond = 1;
t0 = Tp; Xsc_0 = SUN2PLANET(Tp-states(1,1)*86400, X(end,:),'Uranus');
tf = t0 + 1200*86400;

[T,X] = SCPROP(t0,tf, Xsc_0, opts0); Tp = T; T = SEC2JULIAN(T-states(1,1)*86400);
urafly = X; X = PLANET2SUN(Tp-states(1,1)*86400, X, 'Uranus'); Tp = T(end);

Xsc_tot = [Xsc_tot;X]; tlin = [tlin, T'];
grav3 = norm(X(end,4:6))- norm(X(1,4:6)); gravtot = grav3+gravtot;
%% Uranus to Neptune Segment
opts0.cBody = 'Sun'; opts0.tBody = 'Neptune'; opts0.stopCond = 0;
t0 = Tp; Xsc_0 = X(end,:)';
tf = states(5,1); rf = states(5, 2:4);

[v0,delv4] = calcburn(Xsc_0, rf, tf-t0, opts0); Xsc_0 = [Xsc_0(1:3); v0];
[T,X] = SCPROP(t0*86400, tf*86400, Xsc_0, opts0); Tp = T(end); T = ...
    SEC2JULIAN(T-states(1,1)*86400);

Xsc_tot = [Xsc_tot;X]; tlin = [tlin, T']; delv = delv + delv4;
%% Neptune Fly-By
clear opts0 fig
opts0.cBody = 'Neptune'; opts0.stopCond = 1;
t0 = Tp; Xsc_0 = SUN2PLANET(Tp-states(1,1)*86400, X(end,:),'Neptune');
tf = t0 + 1200*86400;

[T,X] = SCPROP(t0,tf, Xsc_0, opts0); Tp = T; T = SEC2JULIAN(T-states(1,1)*86400);
nepfly = X; X = PLANET2SUN(Tp-states(1,1)*86400, X, 'Neptune'); Tp = T(end);

Xsc_tot = [Xsc_tot;X]; tlin = [tlin, T'];
grav4 = norm(X(end,4:6))- norm(X(1,4:6)); gravtot = grav4+gravtot;
%% Post Neptune Segment
opts0.cBody = 'Sun'; opts0.tBody = 'Uranus'; opts0.stopCond = 0;
t0 = Tp; Xsc_0 = X(end,:)';
tf = juliandate(1990,1,0,0,0,0);

[T,X] = SCPROP(t0*86400, tf*86400, Xsc_0, opts0); Tp = T(end); T = ...
    SEC2JULIAN(T-states(1,1)*86400);

Xsc_tot = [Xsc_tot;X]; tlin = [tlin, T'];
%% Mission Analysis - Correction Burns
fprintf('\n\n\tVoyager 2 Mission Analysis - Burns')
fprintf('\n   ======================================')
fprintf('\n\tEarth Correction Burn:   %.3f [km/s]',delv1)
fprintf('\n\tJupiter Correction Burn: %.3f [km/s]',delv2)
fprintf('\n\tSaturn Correction Burn:  %.3f [km/s]',delv3)
fprintf('\n\tUranus Correction Burn:  %.3f [km/s]',delv4)
fprintf('\n\tTotal Burn:              %.3f [km/s]',delv)

%% Mission Analysis - Delta V's
fprintf('\n\n\tVoyager 2 Mission Analysis - Assists')
fprintf('\n   ======================================')
fprintf('\n\tJupiter Gravity Assist: %.3f [km/s]',grav1)
fprintf('\n\tSaturn Gravity Assist:  %.3f [km/s]',grav2)
fprintf('\n\tUranus Gravity Assist:  %.3f [km/s]',grav3)
fprintf('\n\tNeptune Gravity Assist: %.3f [km/s]',grav4)
fprintf('\n\tTotal Velocity Assist:  %.3f [km/s]',gravtot)

%% Plot Trajectory
%% Helio-Centric
```

```
fig.cBody = 'Sun';
fig.tBody = 'Earth'; Xe_tot = PLANETLOC(tlin',fig)';
fig.tBody = 'Jupiter'; Xjup_tot = PLANETLOC(tlin',fig)';
fig.tBody = 'Saturn'; Xsat_tot = PLANETLOC(tlin',fig)';
fig.tBody = 'Uranus'; Xura_tot = PLANETLOC(tlin',fig)';
fig.tBody = 'Neptune'; Xnep_tot = PLANETLOC(tlin',fig)';

figure()
hold on
plot3(Xsc_tot(:,1), Xsc_tot(:,2), Xsc_tot(:,3),'k', 'linewidth', 1.5)
plot3(Xe_tot(:,1), Xe_tot(:,2), Xe_tot(:,3),'color',[0.2,0.5,0.2], 'linewidth', 2)
plot3(Xjup_tot(:,1), Xjup_tot(:,2), Xjup_tot(:,3),'color', [0.8,.2,0], 'linewidth', 2)
plot3(Xsat_tot(:,1), Xsat_tot(:,2), Xsat_tot(:,3), 'color', [0.85,0.85,0], 'linewidth', 2)
plot3(Xura_tot(:,1), Xura_tot(:,2), Xura_tot(:,3), 'color', [0,0.85,0.85], 'linewidth', 2)
plot3(Xnep_tot(:,1), Xnep_tot(:,2), Xnep_tot(:,3), 'color', [0.1,0.1,0.8], 'linewidth', 2)
scatter3(states(:,2),states(:,3),states(:,4),'k')
legend({['Voyager 2',newline,' ...
    Trajectory'],'Earth','Jupiter','Saturn','Uranus','Neptune'},'interpreter', 'latex', ...
    'fontsize', 18)
title({'Voyager 2 Heliocentric','Mission Overview'},'interpreter','latex','fontsize', 18)
axis equal; axis tight
grid on
xlabel('X-Axis, $km$', 'interpreter', 'latex', 'fontsize',18)
ylabel('Y-Axis, $km$', 'interpreter', 'latex', 'fontsize',18)
zlabel('Z-Axis, $km$', 'interpreter', 'latex', 'fontsize',18)
%% Earth Centered
fig.cBody = 'Earth';
Xsc_tote = SUN2PLANETdan(tlin', Xsc_tot, 'Earth');

fig.tBody = 'Jupiter'; jup = PLANETLOC(tlin',fig)';
Xjup_tote = PLANET2SUN(tlin', jup, 'Earth');

fig.tBody = 'Saturn'; sat = PLANETLOC(tlin',fig)';
Xsat_tote = PLANET2SUN(tlin', sat, 'Earth');

fig.tBody = 'Uranus'; ura = PLANETLOC(tlin',fig)';
Xura_tote = PLANET2SUN(tlin', ura, 'Earth');

fig.tBody = 'Neptune'; nep = PLANETLOC(tlin',fig)';
Xnep_tote = PLANET2SUN(tlin', nep, 'Earth');

figure()
hold on
plot3(Xsc_tote(:,1), Xsc_tote(:,2), Xsc_tote(:,3),'k', 'linewidth', 1.5)
plot3(Xjup_tote(:,1), Xjup_tote(:,2), Xjup_tote(:,3),'color', [0.8,.2,0], 'linewidth', 2)
plot3(Xsat_tote(:,1), Xsat_tote(:,2), Xsat_tote(:,3), 'color', [0.85,0.85,0], 'linewidth', 2)
plot3(Xura_tote(:,1), Xura_tote(:,2), Xura_tote(:,3), 'color', [0,0.85,0.85], 'linewidth', 2)
plot3(Xnep_tote(:,1), Xnep_tote(:,2), Xnep_tote(:,3), 'color', [0.1,0.1,0.8], 'linewidth', 2)
legend({['Voyager 2',newline,' ...
    Trajectory'],'Jupiter','Saturn','Uranus','Neptune'},'interpreter', 'latex', ...
    'fontsize', 18)
title({'Voyager 2 Fly-bys','Earth-Centered'},'interpreter','latex','fontsize', 18)
axis equal; axis tight
grid on
xlabel('X-Axis, $km$', 'interpreter', 'latex', 'fontsize',18)
ylabel('Y-Axis, $km$', 'interpreter', 'latex', 'fontsize',18)
zlabel('Z-Axis, $km$', 'interpreter', 'latex', 'fontsize',18)
%% Jupiter Flyby
figure()
hold on
plot3(jupfly(:,1), jupfly(:,2), jupfly(:,3),'k', 'linewidth', 1.5)
plotplanet(71492, [0.8,.2,0])
legend({['Voyager 2',newline,' Trajectory'],'Jupiter'},'interpreter', 'latex', 'fontsize', ...
    18, 'location', 'southwest')
title({'Voyager 2 Flyby of Jupiter'},'interpreter','latex','fontsize', 18)
axis equal;
xlabel('X-Axis, $km$', 'interpreter', 'latex', 'fontsize',18)
ylabel('Y-Axis, $km$', 'interpreter', 'latex', 'fontsize',18)
```

```
zlabel('Z-Axis, $km$', 'interpreter', 'latex', 'fontsize',18)

figure()
hold on
plot3(jupfly(250:400,1), jupfly(250:400,2), jupfly(250:400,3),'k', 'linewidth', 1.5)
plotplanet(71492, [0.8,.2,0])
legend({['Voyager 2',newline,' Trajectory'],'Jupiter'},'interpreter', 'latex', 'fontsize', 18)
title({'Voyager 2 Flyby of Jupiter'},'interpreter','latex','fontsize', 18)
axis equal;
xlabel('X-Axis, $km$', 'interpreter', 'latex', 'fontsize',18)
ylabel('Y-Axis, $km$', 'interpreter', 'latex', 'fontsize',18)
zlabel('Z-Axis, $km$', 'interpreter', 'latex', 'fontsize',18)
%% Saturn Flyby
figure()
hold on
plot3(satfly(:,1), satfly(:,2), satfly(:,3),'k', 'linewidth', 1.5)
plotplanet(58232, [0.85,0.85,0])
legend({['Voyager 2',newline,' Trajectory'],'Jupiter'},'interpreter', 'latex', 'fontsize', 18)
title({'Voyager 2 Flyby of Saturn'},'interpreter','latex','fontsize', 18)
axis equal;
xlabel('X-Axis, $km$', 'interpreter', 'latex', 'fontsize',18)
ylabel('Y-Axis, $km$', 'interpreter', 'latex', 'fontsize',18)
zlabel('Z-Axis, $km$', 'interpreter', 'latex', 'fontsize',18)

figure()
hold on
plot3(satfly(10:400,1), satfly(10:400,2), satfly(10:400,3),'k', 'linewidth', 1.5)
plotplanet(58232, [0.85,0.85,0])
legend({['Voyager 2',newline,' Trajectory'],'Jupiter'},'interpreter', 'latex', 'fontsize', ...
    18, 'location', 'southeast')
title({'Voyager 2 Flyby of Saturn'},'interpreter','latex','fontsize', 18)
axis equal;
xlabel('X-Axis, $km$', 'interpreter', 'latex', 'fontsize',18)
ylabel('Y-Axis, $km$', 'interpreter', 'latex', 'fontsize',18)
zlabel('Z-Axis, $km$', 'interpreter', 'latex', 'fontsize',18)
%% Uranus Flyby
figure()
hold on
plot3(urafly(:,1), urafly(:,2), urafly(:,3),'k', 'linewidth', 1.5)
plotplanet(25361, [0,0.85,0.85])
legend({['Voyager 2',newline,' Trajectory'],'Jupiter'},'interpreter', 'latex', 'fontsize', 18)
title({'Voyager 2 Flyby of Uranus'},'interpreter','latex','fontsize', 18)
axis equal;
xlabel('X-Axis, $km$', 'interpreter', 'latex', 'fontsize',18)
ylabel('Y-Axis, $km$', 'interpreter', 'latex', 'fontsize',18)
zlabel('Z-Axis, $km$', 'interpreter', 'latex', 'fontsize',18)

figure()
hold on
plot3(urafly(10:200,1), urafly(10:200,2), urafly(10:200,3),'k', 'linewidth', 1.5)
plotplanet(25361, [0,0.85,0.85])
legend({['Voyager 2',newline,' Trajectory'],'Jupiter'},'interpreter', 'latex', 'fontsize', ...
    18, 'location', 'southeast')
title({'Voyager 2 Flyby of Uranus'},'interpreter','latex','fontsize', 18)
axis equal;
xlabel('X-Axis, $km$', 'interpreter', 'latex', 'fontsize',18)
ylabel('Y-Axis, $km$', 'interpreter', 'latex', 'fontsize',18)
zlabel('Z-Axis, $km$', 'interpreter', 'latex', 'fontsize',18)
%% Neptune Flyby
figure()
hold on
plot3(nepfly(:,1), nepfly(:,2), nepfly(:,3),'k', 'linewidth', 1.5)
plotplanet(24621 , [0.1,0.1,0.8])
legend({['Voyager 2',newline,' Trajectory'],'Jupiter'},'interpreter', 'latex', 'fontsize', 18)
title({'Voyager 2 Flyby of Neptune'},'interpreter','latex','fontsize', 18)
axis equal;
xlabel('X-Axis, $km$', 'interpreter', 'latex', 'fontsize',18)
ylabel('Y-Axis, $km$', 'interpreter', 'latex', 'fontsize',18)
```

```
zlabel('Z-Axis, $km$', 'interpreter', 'latex', 'fontsize',18)

figure()
hold on
plot3(nepfly(10:200,1), nepfly(10:200,2), nepfly(10:200,3),'k', 'linewidth', 1.5)
plotplanet(24621, [0.1,0.1,0.8])
legend({['Voyager 2',newline,' Trajectory'],'Jupiter'},'interpreter', 'latex', 'fontsize', ...
    18, 'location', 'southwest')
title({'Voyager 2 Flyby of Neptune'},'interpreter','latex','fontsize', 18)
axis equal;
xlabel('X-Axis, $km$', 'interpreter', 'latex', 'fontsize',18)
ylabel('Y-Axis, $km$', 'interpreter', 'latex', 'fontsize',18)
zlabel('Z-Axis, $km$', 'interpreter', 'latex', 'fontsize',18)
%% Velocity Profile
rnorm = sqrt(Xsc_tot(:,1).^2 + Xsc_tot(:,2).^2 +Xsc_tot(:,3).^2 );
vesc = sqrt(2*132712000000./rnorm);
vnorm = sqrt(Xsc_tot(:,4).^2 + Xsc_tot(:,5).^2 +Xsc_tot(:,6).^2 );

figure()
hold on
plot(rnorm./149597871, vnorm,'k', 'linewidth', 1.8)
plot(rnorm./149597871, vesc, 'k--', 'linewidth', 1.8)
xlabel('Distance from Sun, $Au$','interpreter', 'latex', 'fontsize', 18)
ylabel('Velocity, $\frac{km}{s}$','interpreter', 'latex', 'fontsize', 18)
title({'Escape Velocity From','Solar System'},'interpreter', 'latex', 'fontsize', 18)
legend({['Voyager 2',newline,'Velocity Profile'],['Escape Velocity',newline,'From Sun ...
    $V_{esc}$']},'interpreter', 'latex', 'fontsize', 18)
%% Additional Help Functions
function [an, delv] = calcburn(X, XT, dt, opts0)
    dt = dt * 86400;
    [V1, ¬] = LAMBERTSOLVE(X(1:3,1), XT(1:3)', dt, opts0);

    if norm(abs(V1 - X(4:end,1)))>1E-7
        an = V1;
        delv = abs(norm(V1) - norm(X(4:end,1)));
    else
        an =  X(4:end,1);
        delv = 0;
    end
end
function plotplanet(rad, col)
num = 25; phi = linspace(0, 2*pi, num);
lambda = linspace(0, 2*pi, num);

    for i = 1:num
        x = rad.*cos(lambda).*sin(phi(i));
        y = rad.*sin(lambda).*sin(phi(i));
        z = rad.*cos(phi(i)).*ones(num);

        x2 = rad.*cos(lambda(i)).*sin(phi);
        y2 = rad.*sin(lambda(i)).*sin(phi);
        z2 = rad.*cos(phi);

        plot3(x, y, z, 'color', col, 'linewidth', 1.8)
        plot3(x2, y2, z2, 'color', col, 'linewidth', 1.8)

    end
end
```

Command Window:
Voyager 2 Mission Analysis - Burns
========================
Earth Correction Burn: 0.095 [km/s]
Jupiter Correction Burn: 0.361 [km/s]
Saturn Correction Burn: 2.362 [km/s]
Uranus Correction Burn: 1.487 [km/s]
Total Burn: 4.305 [km/s]

Voyager 2 Mission Analysis - Assists
========================
Jupiter Gravity Assist: 8.262 [km/s]
Saturn Gravity Assist: 2.478 [km/s]
Uranus Gravity Assist: 0.281 [km/s]
Neptune Gravity Assist: 0.493 [km/s]
Total Velocity Assist: 11.513 [km/s]

B Lamberts Problem

J. H. Lambert (1728–1777) was a French-born German astronomer, physicist, and mathematician. Lambert proposed that the transfer time Δt from P_1 to P_2 is independent of the orbit's eccentricity and depends only on the sum $r_1 + r_2$ of the magnitudes of the position vectors, the semimajor axis a and the length c of the chord joining P_1 and P_2. It is noteworthy that the period (of an ellipse) and the specific mechanical energy are also independent of the eccentricity.[2]

$$\vec{r}_2 = f\vec{r}_1 + g\vec{v}_1 \tag{3}$$
$$\vec{v}_2 = \dot{f}\vec{r}_1 + \dot{g}\vec{v}_1 \tag{4}$$

Where $f, \dot{f}, g, \dot{g}$ is defined from

$$f = 1 - \frac{y(z)}{r_1} \tag{5}$$
$$g = A\sqrt{y(z)} \tag{6}$$
$$\dot{f} = \frac{\sqrt{\mu}}{r_1 r_2}\sqrt{\frac{y(z)}{C(z)}}[zS(z) - 1] \tag{7}$$
$$\dot{g} = 1 - \frac{y(z)}{r_2} \tag{8}$$

References

[1] N. JPL, "Mission to jupiter, saturn, uranus, neptune voyager 2."

[2] H. D. Curtis, Orbital Mechanics for Engineering Students.