HW9 Problem 1

```
fprintf('\n');
clearvars -except function_list hw_pub toolsPath
close all
CelestialConstants; % import useful constants

h_earth = 185; %km
h_mars = 300; %km

r_soi_earth = Earth.a*(Earth.m/Sun.m)^(2/5)
r_soi_mars = Mars.a*(Mars.m/Sun.m)^(2/5)
r_soi_Earth_Moon = Earth.a*((Moon.m+Earth.m)/Sun.m)^(2/5)

% Transfer properties
a_xfer = (Earth.a + Mars.a)/2;
v_xfer_i = sqrt(2*Sun.mu/Earth.a - Sun.mu/a_xfer);
v_xfer_f = sqrt(2*Sun.mu/Mars.a - Sun.mu/a_xfer);

v_earth = sqrt(Sun.mu/Earth.a);
v_mars = sqrt(Sun.mu/Mars.a);

v_inf_earth = abs(v_xfer_i-v_earth);
v_inf_mars = abs(v_mars-v_xfer_f);

v_park_earth = sqrt(Earth.mu/(Earth.R+h_earth));
v_park_mars = sqrt(Mars.mu/(Mars.R+h_mars));

dv_earth_inj = sqrt(2*Earth.mu/(Earth.R+h_earth)+v_inf_earth^2)...
    -v_park_earth;
dv_mars_ins = sqrt(2*Mars.mu/(Mars.R+h_mars)+v_inf_mars^2)-v_park_mars;

T_xfer = 2*pi*sqrt(a_xfer^3/Sun.mu)/2/3600/24;


fprintf('a) Earth SOI: %.0f km\n',r_soi_earth)
fprintf('   Mars SOI: %.0f km\n',r_soi_mars)
fprintf('   Earth-Moon SOI: %.0f km\n',r_soi_Earth_Moon)
fprintf('b) Earth heliocentric departure velocity: %.3f km/s\n',v_xfer_i)
fprintf('   Mars heliocentric arrival velocity: %.3f km/s\n',v_xfer_f)
fprintf('c) Earth departure v_inf: %.3f km/s\n',v_inf_earth)
fprintf('   Mars arrival v_inf: %.3f km/s\n',v_inf_mars)
fprintf('d) Mars transfer injection dv: %.3f km/s\n',dv_earth_inj)
fprintf('e) Mars insertion dv: %.3f km/s\n',dv_mars_ins)
fprintf('f) Transfer time: %.1f days\n',T_xfer)
```

```
r_soi_earth =

   9.2465e+05
```

```
r_soi_mars =

   5.7723e+05


r_soi_Earth_Moon =

   9.2918e+05

a) Earth SOI: 924645 km
   Mars SOI: 577226 km
   Earth-Moon SOI: 929178 km
b) Earth heliocentric departure velocity: 32.729 km/s
   Mars heliocentric arrival velocity: 21.480 km/s
c) Earth departure v_inf: 2.945 km/s
   Mars arrival v_inf: 2.649 km/s
d) Mars transfer injection dv: 3.615 km/s
e) Mars insertion dv: 2.093 km/s
f) Transfer time: 258.9 days
```

Published with MATLAB® R2013b

HW9 Problem 2

```
fprintf('\n');
clearvars -except function_list hw_pub toolsPath
close all
CelestialConstants; % import useful constants

rp = 200+Mars.R;

% Transfer properties
a_xfer = (Earth.a + Mars.a)/2;
v_xfer_f = sqrt(2*Sun.mu/Mars.a - Sun.mu/a_xfer);

v_mars = sqrt(Sun.mu/Mars.a);
v_inf_mars = abs(v_mars-v_xfer_f);

v_assist_mars_max = abs(v_inf_mars + v_mars);
v_assist_mars_min = abs(v_inf_mars - v_mars);

fprintf(['Minimum heliocentric velocity from Mars gravity assist: %.3f'...
    ' km/s\n'],v_assist_mars_min)
fprintf(['Maximum heliocentric velocity from Mars gravity assist: %.3f'...
    ' km/s\n'],v_assist_mars_max)
```

```
Minimum heliocentric velocity from Mars gravity assist: 21.480 km/s
Maximum heliocentric velocity from Mars gravity assist: 26.778 km/s
```

Published with MATLAB® R2013b

HW9 Problem 3

```
fprintf('\n');
clearvars -except function_list hw_pub toolsPath
close all
CelestialConstants; % import useful constants

rp = Jupiter.R*2;

% Transfer properties
a_xfer = (Earth.a + Jupiter.a)/2;
v_xfer_f = sqrt(2*Sun.mu/Jupiter.a - Sun.mu/a_xfer);

v_jup = sqrt(Sun.mu/Jupiter.a);
v_inf_jup = abs(v_jup-v_xfer_f);

v_assist_jup_max = abs(v_inf_jup + v_jup);

v_esc_jup = sqrt(2*Sun.mu/Jupiter.a);

spec_energy = v_assist_jup_max^2/2-Sun.mu/Jupiter.a; %J/kg

fprintf('Max velocity from gravity assist: %.3f km/s\n', v_assist_jup_max)
fprintf('Heliocentric escape velocity at Jupiter: %.3f km/s\n', v_esc_jup)
fprintf('The spacecraft can escape the solar system in this mission plan.\n')
fprintf('Max specific energy: %.3f J/kg\n', spec_energy)
```

```
Max velocity from gravity assist: 18.701 km/s
Heliocentric escape velocity at Jupiter: 18.467 km/s
The spacecraft can escape the solar system in this mission plan.
Max specific energy: 4.354 J/kg
```

Published with MATLAB® R2013b

HW9 Problem 4

```
fprintf('\n');
clearvars -except function_list hw_pub toolsPath
close all
CelestialConstants; % import useful constants


r_soi_jupiter = Jupiter.a*(Jupiter.m/Sun.m)^(2/5);
jup_stable_orbit = 0.67*r_soi_jupiter;

r_soi_saturn = Saturn.a*(Saturn.m/Sun.m)^(2/5);
sat_stable_orbit = 0.67*r_soi_saturn;

fprintf(['S/2003 J 2 is within the stable region of Jupiter\n(' ...
    num2str(jup_stable_orbit,'%.0f') ' km), so it is likely to be \n'...
    'stable. Fornjot is within the stable region of Saturn\n(' ...
    num2str(sat_stable_orbit,'%.0f') ' km), so it is also likely to be \n'...
    'stable. Since these satellites are well-within the stable regions,\n'...
    'it is doubtful additional satellites will be discovered outside \n'...
    'this limit. However, other satellites could be found in the same \n'...
    'orbital vicinity.\n'])
```

```
S/2003 J 2 is within the stable region of Jupiter
(32296091 km), so it is likely to be
stable. Fornjot is within the stable region of Saturn
(36614772 km), so it is also likely to be
stable. Since these satellites are well-within the stable regions,
it is doubtful additional satellites will be discovered outside
this limit. However, other satellites could be found in the same
orbital vicinity.
```

Published with MATLAB® R2013b

CelestialConstants

Table of Contents

Description

All sorts of constants for orbital mechanics purposes

```
fcnPrintQueue(mfilename('fullpath')) % Add this code to code app
```

Earth

```
Earth.name = 'Earth';
Earth.mu = 3.986004415e5; %km3/s2
Earth.R = 6378; %km
Earth.a = 149598023; %km
Earth.spin_rate = 7.2921158553e-05; %rad/s
Earth.flattening = 1/298.25722; %WGS-84
Earth.J2 = 0.0010826267;
Earth.P_days = 365.2421897; %days
Earth.P_years = 0.99997862; %days
Earth.m = 5.9742e24; %kg
```

Moon

```
Moon.name = 'Moon';
Moon.R = 1738.0; %km
Moon.J2 = 0.0002027;
Moon.P_days = 27.321582; %days
Moon.mu = 4902.799; %km3/s2
Moon.m = 7.3483e22; %kg
Moon.a = 384400; %km
```

Sun

```
Sun.mu = 1.32712428e11; %km3/s2
Sun.m = 1.9891e30; %kg
```

Mercury

```
Mercury.name = 'Mercury';
Mercury.R = 2439.0; %km
Mercury.J2 = 0.00006;
Mercury.P_days = 87.9666; %days
Mercury.mu = 2.2032e4; %km3/s2
```

Venus

```
Venus.name = 'Venus';
Venus.a = 108208601; %km
Venus.R = 6052.0; %km
Venus.J2 = 0.000027;
Venus.P_days = 224.6906; %days
Venus.mu = 3.257e5; %km3/s2
```

Mars

```
Mars.name = 'Mars';
Mars.a = 227939186; %km
Mars.R = 3397.2; %km
Mars.J2 = 0.001964;
Mars.P_days = 686.9150; %days
Mars.mu = 4.305e4; %km3/s2
Mars.m = 6.4191e23; %kg
```

Jupiter

```
Jupiter.name = 'Jupiter';
Jupiter.a = 778298361; %km
Jupiter.R = 71492; %km
Jupiter.J2 = 0.01475;
Jupiter.P_years = 11.856525; %days
Jupiter.P_days = Jupiter.P_years/Earth.P_years*Earth.P_days; %days
Jupiter.mu = 1.268e8; %km3/s2
Jupiter.m = 1.8988e27; %kg
```

Saturn

```
Saturn.name = 'Saturn';
Saturn.a = 1429394133; %km
Saturn.R = 60268; %km
Saturn.J2 = 0.01645;
```

```
Saturn.P_years = 29.423519; %days
Saturn.P_days = Saturn.P_years/Earth.P_years*Earth.P_days; %days
Saturn.mu = 3.794e7; %km3/s2
Saturn.m = 5.685e26; %kg
```

Uranus

```
Uranus.name = 'Uranus';
Uranus.R = 25559; %km
Uranus.J2 = 0.012;
Uranus.P_years = 83.747406; %days
Uranus.P_days = Uranus.P_years/Earth.P_years*Earth.P_days; %days
Uranus.mu = 5.794e6; %km3/s2
```

Neptune

```
Neptune.name = 'Neptune';
Neptune.R = 24764; %km
Neptune.J2 = 0.004;
Neptune.P_years = 163.7232045; %days
Neptune.P_days = Neptune.P_years/Earth.P_years*Earth.P_days; %days
Neptune.mu = 6.809e6; %km3/s2
```

Celestial units

```
au2km = 149597870.7;
```

Physical constants

```
day2sec = 86400; % sec/day
speed_of_light = 299792458; %m/s
```

Published with MATLAB® R2013b

```
function fcnPrintQueue( filename )
global function_list;
if exist('function_list', 'var')
    file_in_list = 0;
    for idx = 1:length(function_list)
        if strcmp(function_list(idx), filename);
            file_in_list = 1;
            break
        end
    end
    if ~file_in_list
        function_list = [function_list, filename];
    end
end
end
```

Published with MATLAB® R2013b