HW2 Problem 1

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

e = 0.15;
a = 2*6378; % km
mu = 3.986e5; % km3/s2
n = sqrt(mu/a/a/a);
d2r = pi/180;

cases = {'A'; 'B'; 'C'; 'D'; 'E'; 'F'};
f = {'0.00'};
E = {'0.00'};
M = {'0.00'};
T = {'0.00'};

f = [f; '30.00'];
E = [E; num2str(f2E(30*d2r,e)/d2r, '%.2f')];
M = [M; num2str(E2M(f2E(30*d2r,e),e)/d2r, '%.2f')];
T = [T; num2str(E2M(f2E(30*d2r,e),e)/n/60, '%.2f')];

f = [f; num2str(E2f(200*d2r,e)/d2r, '%.2f')];
E = [E; '200.00'];
M = [M; num2str(E2M(200*d2r,e)/d2r, '%.2f')];
T = [T; num2str(E2M(200*d2r,e)/n/60, '%.2f')];

f = [f; '90.00'];
E = [E; num2str(f2E(90*d2r,e)/d2r, '%.2f')];
M = [M; num2str(E2M(f2E(90*d2r,e),e)/d2r, '%.2f')];
T = [T; num2str(E2M(f2E(90*d2r,e),e)/n/60, '%.2f')];

f = [f; num2str(E2f(M2E(270*d2r,e),e)/d2r, '%.2f')];
E = [E; num2str(M2E(270*d2r,e)/d2r, '%.2f')];
M = [M; '270.00'];
T = [T; num2str(270*d2r/n/60, '%.2f')];

f = [f; num2str(E2f(E2M(n*25*60,e),e)/d2r, '%.2f')];
E = [E; num2str(E2M(n*25*60,e)/d2r, '%.2f')];
M = [M; num2str(n*25*60/d2r, '%.2f')];
T = [T; '25.00'];


imshow(imread('HW2_P1_graphic1.jpg'));
% disp(table(cases, f, E, M, T))%, 'VariableNames', ...
%     {'Case', 'True Anom (deg)', 'Eccentric Anom (deg)', ...
%     'Mean Anom (deg)', 'T-Tp (min)'}))
% f
% E
% M
```

```
% T
```

Case	True Anom (deg)	Eccentric Anom (deg)	Mean Anom (deg)	Time Since Periapse (min)
A	0	0	0	0
B	30	25.95	22.18	14.73
C	197.24	200	202.94	134.71
D	90	81.37	72.88	48.37
E	253.06	261.5	270	179.22
F	37.36	32.41	37.66	25

Published with MATLAB® R2013b

HW2 Problem 2

```
fprintf('\n');
clearvars -except function_list hw_pub toolsPath
close all

hp = 321; %km
ha = 551; %km
f = 330*pi/180; %rad
delta_t = 65*60; %min
Re = 6378; %km
mu = 3.986e5; %km3/s2

a = (hp + Re + Re + ha)/2;
e = (a-hp-Re)/a;
n = sqrt(mu/a/a/a);
M0 = E2M(f2E(f,e),e);
M_deploy = M0 + n*delta_t;
if M_deploy > 2*pi
    M_deploy = M_deploy - 2*pi;
end

f_deploy = E2f(M2E(M_deploy,e), e);
fprintf('Shuttle true anom at satellite deployment: %.2f deg\n',...
    f_deploy*180/pi)
```

```
Shuttle true anom at satellite deployment: 220.50 deg
```

Published with MATLAB® R2013b

HW2 Problem 3

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

hp_actual = 798; %km
ha_actual = 817; %km
hp_planned = 794; %km
ha_planned = 814; %km

a_actual = (hp_actual + ha_actual + 2*Earth.R)/2;
a_planned = (hp_planned + ha_planned + 2*Earth.R)/2;

e_actual = 1-(hp_actual+Earth.R)/a_actual;
e_planned = 1-(hp_planned+Earth.R)/a_planned;

P_actual = 2*pi*sqrt(a_actual*a_actual*a_actual/Earth.mu);
P_planned = 2*pi*sqrt(a_planned*a_planned*a_planned/Earth.mu);

fprintf('Semimajor axis:\n')
fprintf('\t%.2f km difference\n', abs(a_planned-a_actual))
fprintf('\t%.2f%% error\n\n', abs(a_planned-a_actual)/a_planned*100)
fprintf('Eccentricity:\n')
fprintf('\t%.6f difference\n', abs(e_planned-e_actual))
fprintf('\t%.2f%% error\n\n', abs(e_planned-e_actual)/e_planned*100)
fprintf('Period:\n')
fprintf('\t%.2f s difference\n', abs(P_planned-P_actual))
fprintf('\t%.2f%% error\n\n', abs(P_planned-P_actual)/P_planned*100)
```

```
Semimajor axis:
 3.50 km difference
 0.05% error

Eccentricity:
 0.000070 difference
 5.05% error

Period:
 4.43 s difference
 0.07% error
```

Published with MATLAB® R2013b

HW2 Problem 4

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

r = [-5650;-2650;2850]; %km
v = [2.415;-7.032;-1.796]; %km/s

[a,e,i,RAAN,w,f] = cart2OE(r,v,Earth.mu);

rp = a*(1-e);
hp = rp - Earth.R;
ra = a*(1+e);
ha = ra - Earth.R;

specific_energy = norm(v)*norm(v)/2-Earth.mu/norm(r);

h = cross(r,v);

fpa = atan2(e*sin(f),1+e*cos(f));

fprintf('a. Semi-major axis: %.2f km\n', a)
fprintf('b. Eccentricity: %.2f\n', e)
fprintf('c. Perigee altitude: %.2f km\n', hp)
fprintf('   Apogee altitude: %.2f km\n', ha)
fprintf('d. Specific energy: %.2f J/kg\n', specific_energy)
fprintf('e. Angular momentum: %.2f km2/s\n', h(1))
fprintf('                     %.2f km2/s\n', h(2))
fprintf('                     %.2f km2/s\n', h(3))
fprintf('f. Inclination: %.2f deg\n', i*180/pi)
fprintf('b. Flight path angle: %.2f deg\n', fpa*180/pi)
```

```
a. Semi-major axis: 6908.95 km
b. Eccentricity: 0.01
c. Perigee altitude: 479.70 km
   Apogee altitude: 582.20 km
d. Specific energy: -28.85 J/kg
e. Angular momentum: 24800.60 km2/s
                     -3264.65 km2/s
                     46130.55 km2/s
f. Inclination: 28.47 deg
b. Flight path angle: -0.14 deg
```

Published with MATLAB® R2013b

HW2 Problem 5

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

r = 8200; %km
vcirc = sqrt(Earth.mu/r);

fprintf('Satellite speed should be %.3f km/s\n', vcirc)
fprintf('Satellite FPA should be 0 degrees\n')
```

```
Satellite speed should be 6.972 km/s
Satellite FPA should be 0 degrees
```

Published with MATLAB® R2013b

HW2 Problem 6

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

a = 0.387; %AU
e = 0.205;
TU2days = 365.25/(2*pi);

P = 2*pi*sqrt(a*a*a)*TU2days; %s
ra = a*(1+e);
rp = a*(1-e);

vp = sqrt(2/rp - 1/a)*au2km/TU2days/day2sec;
fprintf('Aphelion = %.3f AU\n',ra);
fprintf('Perihelion = %.3f AU\n',rp);
fprintf('Perihelion speed = %.1f km/s\n',vp);
```

```
Aphelion = 0.466 AU
Perihelion = 0.308 AU
Perihelion speed = 58.9 km/s
```

Published with MATLAB® R2013b

CelestialConstants

Table of Contents

Description 1
Earth 1
Celestial units 1
Physical constants 1

Description

All sorts of constants for orbital mechanics purposes

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

Earth

```
Earth.mu = 3.986e5; %km3/s2
Earth.R = 6378; %km
```

Celestial units

```
au2km = 149597870.7;
```

Physical constants

```
day2sec = 86400;
```

Published with MATLAB® R2013b

```
function E = f2E( f, e )
%f2E True anomaly (f) to eccentric anomaly (E)
% Only valid for e < 1
% units in radians
fcnPrintQueue(mfilename('fullpath')) % Add this code to code app

% Vallado eqn 2-9
E = acos((e + cos(f))/(1+e*cos(f)));
if f > pi
    E = 2*pi - E;
end
```

Published with MATLAB® R2013b

```
function M = E2M( E, e )
%E2M Eccentric anomaly (E) to mean anomaly (M)
% units in radians
fcnPrintQueue(mfilename('fullpath')) % Add this code to code app

% Vallado eqn 2-4
M = E-e*sin(E);
```

Published with MATLAB® R2013b

```
function f = E2f( E, e )
%E2f Eccentric anomaly (E) to true anomaly (f)
% Only valid for e < 1
% units in radians
fcnPrintQueue(mfilename('fullpath')) % Add this code to code app

% Vallado eqn 2-10
f = acos((cos(E) - e)/(1-e*cos(E)));
if E > pi
    f = 2*pi - f;
end
```

Published with MATLAB® R2013b

```
function E = M2E( M, e )
%M2E Mean anom (M) to eccentric anom (E)
fcnPrintQueue(mfilename('fullpath')) % Add this code to code app

tol = 1e-5;

if (M < 0 && M > -pi) || M > pi
    E_1 = M - e;
else
    E_1 = M + e;
end

E = E_1 + tol + 1;
while abs(E_1-E) > tol
    E = E_1;
    E_1 = E - (E - e*sin(E) - M)/(1 - e*cos(E));
end
```

Published with MATLAB® R2013b

```
function [a,e,i,RAAN,w,f] = cart2OE( r, v ,mu)
%cart2OE return classical orbital elements from cartesian coords
% Only valid for e < 1
% units in radians
fcnPrintQueue(mfilename('fullpath')) % Add this code to code app

h = cross(r,v);
n = cross([0;0;1],h);
ecc_vec = ((norm(v)*norm(v)-mu/norm(r))*r - dot(r,v)*v)/mu;

e = norm(ecc_vec);
a = 0;
if e < 1.0
    specific_energy = norm(v)*norm(v)/2-mu/norm(r);
    a = -mu/2/specific_energy;
end

i = acos(h(3)/norm(h));
RAAN = acos(n(1)/norm(n));
if n(2) < 0
    RAAN = 2*pi-RAAN;
end
w = acos(dot(n,ecc_vec)/(norm(n)*norm(ecc_vec)));
if ecc_vec(3) < 0
    w = 2*pi-w;
end
f = acos(dot(ecc_vec,r)/(norm(ecc_vec)*norm(r)));
if dot(r,v) < 0
    f = 2*pi-f;
end
```

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