Homework 2 Answer Key

Problem 1

Given:

Component	Value (meters)
Position X	326151.080726
Position Y	6077471.251787
Position Z	2944583.918767
Velocity X	-7455.178720
Velocity Y	-482.482572
Velocity Z	1910.883434

MATLAB solution:

```
Week 2 Homework Solutions

VECTOR 1
------------------------
r:      326151.080726   6077471.251787   2944583.918767  meters
rd:      -7455.178720      -482.482572      1910.883434 m/sec

h:   1.3034049558e+10 -2.2575636068e+10 4.5151272135e+10 m^2/sec

|h|: 5.2136198243e+10 m^2/sec

p:   6819317.99903984 m

E:   -2.9222906294e+07 m^2/s^2

a:   6819999.99902560 m

TP:     5605.15391191 sec

B:   2.1265080535e+12 3.2207421496e+12 9.9650107635e+11

|B|: 3.9860043767e+12

e:       0.0099999999

nu:      0.5337080028 rad
nu:     30.5792160524 deg
Dot product dot(r, rd) > 0.  Sign is positive.  Quadrant 1 or 2
E0:      0.5286424011 rad
E0:     30.2889784571 deg
M:       0.5235987859 rad
n:       0.0011209657 rad/sec
```

```
dt:    467.0961685125 sec
r65: 6790619.6008190252 m
E65:     1.1254198828 rad
E65:    64.4818094624 deg
dt:    528.8267149214 sec


Kepler Equation Solutions

Echeck:      1.1254198828 rad
Echeck:     64.4818094624 deg
nuCheck:      1.1344640138 rad
nuCheck:     65.0000000000 deg


E2700:      3.5462689773 rad
E2700:    203.1862454165 deg
Perigee crossings after 2700 seconds: 0

nu2700:      3.5423496855 rad
nu2700:    202.9616865415 deg

E2TP:     13.0950130155 rad
E2TP    750.2889784571 deg
Perigee crossings after 1.121031e+04 seconds: 2

In range [0, 2*pi]
E2TP:      0.5286424011 rad
E2TP:     30.2889784571 deg
nu2TP:      0.5337080028 rad
nu2TP:     30.5792160524 deg

E15000:     17.3280965310 rad
E15000:    992.8267982226 deg
Perigee crossings after 15000 seconds: 2

In range [0, 2*pi]
E15000:      4.7617259167 rad
E15000:    272.8267982226 deg
nu15000:      4.7517354548 rad
nu15000:    272.2543869211 deg
```

1. **Find ν_0**

From the Trajectory equation (Vallado Eq. 1-23), the direction of perigee is given by $\vec{B}$, and the true anomaly is computed from

$$\vec{r} \cdot \vec{B} = rB\cos\nu_0$$

Solving for ν_0:

$\vec{B}$ = [2126508042101.570312, 3220742199864.750000, 996501104661.750000]

$\vec{r}$ = [326151.080726   6077471.251787   2944583.918767]

ν_0 =   0.533708 radians

```
               30.579215 degrees
```

e = `0.01`

2. **Find E_0**

$$\sin E_0 = \frac{\sin\nu_0\sqrt{1-e^2}}{(1+e\cos\nu_0)}$$

$$\cos E_0 = \frac{e + \cos\nu_0}{1 + e\cos\nu_0}$$

$$E_0 = \mathrm{atan2}\left(\sin E_0,\ \cos E_0\right) \rightarrow \text{Note use of ATAN2 to return proper quadrant}$$

E_0 =
```
                0.528642 radians
               30.288978 degrees
```

3. **$M_0 = E_0 - e \sin E_0$**

$M_0 = 0.528642 - 0.01 \sin(0.528642)$
M_0 =
```
                0.523599 radians
               30.000000 degrees
```

Find Time of Flight from Perigee to ν_0

$$n = \sqrt{\frac{\mu}{a^3}} = 0.001121 \text{ radians/sec}$$

$n(t-T_0) = n\,\Delta t = M_0$

So

$$\Delta t = \frac{M_0}{n} = 467.096 \text{ sec}$$

Find Time of Flight from ν_0 to $\nu = 65°$

From Vallado (eq 2-7)

$$t - t_0 = \sqrt{\frac{a^3}{\mu}}\left\{2\pi k + \left(E - e\sin E\right) - \left(E_0 - e\sin E_0\right)\right\}$$

Where k is the number of perigee passages between t_0 and t. In this case, k=0.

As above,

$M_0 = E_0 - e \sin E_0$ `0.523599 radians`

Find E in the same manner as above,

cos E	0.430798
sin E	0.902449
E	1.125420 radians
	64.481809 degrees
M = E – e sinE	1.116395 radians
	63.964745 degrees

$$\boxed{t - t_0 = \frac{1}{0.001121}\{0 + 1.116395 - 0.523599\} = 528.826724 \text{ sec}}$$

Kepler Equation Solutions

The Kepler equation is an iterative solution to find the eccentric anomaly after a specified time of flight, starting at an initial eccentric anomaly, E_o. The iteration algorithm is as follows.

Given an initial eccentric anomaly E_o and time of flight Δt, define

$$M_o = E_o - e \sin E_o$$

The general iteration equation is then

$$E_{k+1} = E_k + \frac{n\Delta t + M_0 - (E_k - e \sin E_k)}{1 - e \cos E_k}$$

As written, the iteration will return an eccentric anomaly that is not bounded within the range $[0, 2\pi]$. In Matlab, to express the solved-for value within that range, use the modulo function:

$$E_{bounded} = \text{mod}(E, 2\pi)$$

The number of perigee crossings between E_o and the final time of flight point can be computed in MATLAB as

$$k = \text{floor}(\ (E - E_o) / 2\pi\ )$$

1. Verify: Starting ν_0, compute the true anomaly after the time of flight computed above.

   $\Delta t = 528.8267149214$ sec
   $E_o = 0.5286424011$ rad

   After iteration,

E_{check}: 1.1254198828 rad
E_{check}: 64.4818094624 deg
ν_{check}: 65.0 deg

This is the expected result.

2. Starting at the same E_o, Find the true anomaly after a time of flight of 2700 seconds. After iteration:

   E_{2700}: 3.5462689773 rad
   E_{2700}: 203.1862454165 deg
   nu_{2700}: 3.5423496855 rad
   nu_{2700}: 202.9616865415 deg

3. Starting at the same E_o, Find the true anomaly after exactly two orbit periods?

   E_{2TP}: 13.0950130155 rad
   E_{2TP} 750.2889784571 deg
   Perigee crossings after 1.121031e+04 seconds: 2 (as expected)

   In range [0, 2*pi]
   E_{2TP}: 0.5286424011 rad
   E_{2TP}: 30.2889784571 deg
   $\nu_{2TP`}$: 0.5337080028 rad
   ν_{2TP}: 30.5792160524 deg

   As expected, the results are exactly the same as the initial conditions.

4. Starting at the same E_o, Find the true anomaly after a time of flight of 15000 seconds. Because the Δt for this case exceeds the orbit period, this iteration solution should have multiple perigee crossings.

   E_{15000}: 17.3280965310 rad
   E_{15000}: 992.8267982226 deg
   Perigee crossings after 15000 seconds: 2

   In range [0, 2*pi]
   E_{15000}: 4.7617259167 rad
   E_{15000}: 272.8267982226 deg
   nu_{15000}: 4.7517354548 rad
   nu_{15000}: 272.2543869211 deg

Problem 2

Given:

Component	Value (meters)
Position X	572461.711228
Position Y	-1015437.194396
Position Z	7707337.871302
Velocity X	-6195.262945
Velocity Y	-3575.889650
Velocity Z	-5.423283

1. Find perifocal position and velocity, $\vec{r}_0$ and $\vec{v}_0$

The full set of Keplerian elements for this position and velocity is:

```
a:      7800000.00120126 m
e:            0.00100000
inc:         98.60000000 deg
raan:        30.00000000 deg
wp:          40.00000696 deg
nu:          50.08784582 deg
```

Also, the perifocal position and velocity are computed from the above using,

$$x = r\cos\nu$$
$$y = r\sin\nu$$
$$\dot{x} = -\sqrt{\frac{\mu}{p}}\sin\nu$$
$$\dot{y} = \sqrt{\frac{\mu}{p}}(e + \cos\nu)$$
$$p = a(1 - e^2) = 7799992.2$$

Plugging in the known values, the perifocal position and velocity are given by:

$$\vec{r}_0 = [\,5001362.438709 \quad 5978984.522949 \quad 0.000000]$$
$$\vec{v}_0 = [\,-5483.194150 \quad 4593.787225 \quad 0.000000]$$

Note that the Z-components of position and velocity are zero, as the position and velocity vectors in the perifocal frame are by definition in the orbit plane.

Find f, g, $\dot{f}$, $\dot{g}$ for $\Delta\nu$=33°.

Using the equations from Vallado (2-63)

```
f:                          0.838690
g:                        593.813828
fdot:                      -0.000499
gdot:                       0.838774
```

2. Find $\vec{r}$ and $\vec{v}$ corresponding to Δν

Using:

$$\vec{r} = f\,\vec{r}_0 + g\,\vec{v}_0$$
$$\vec{v} = \dot{f}\,\vec{r}_0 + \dot{g}\,\vec{v}_0$$

$$\vec{r} = [\,938596.059563 \quad 7742368.795881 \quad 0.000000]$$
$$\vec{v} = [\,-7096.655979 \quad 867.465852 \quad 0.000000]$$

Note: you can check your work for this by computing the vectors directly:

Compute $\nu=\nu_0 + \Delta\nu$ `83.087853 degrees`

Compute r using the ellipse equation, then apply the position and velocity equations above.

3. Using energy equation and ellipse equation, find an expression for the orbit speed as a function of true anomaly

Energy equation:

$$\xi = -\frac{\mu}{2a} = \frac{v^2}{2} - \frac{\mu}{r}$$

Rearrange:

$$\frac{\mu}{r} - \frac{\mu}{2a} = \frac{v^2}{2}$$

Multiply through by 2

$$v^2 = \frac{2\mu}{r} - \frac{\mu}{a}$$

Recall from the ellipse equation that

$$r = \frac{a(1-e^2)}{1+e\cos\nu}$$

Substitute into the energy equation for r

$$v^2 = \frac{2\mu(1+e\cos\nu)}{a(1-e^2)} - \frac{\mu}{a}$$

Collect terms

$$v^2 = \frac{\mu}{a}\left(\frac{2+2e\cos\nu}{(1-e^2)} - 1\right)$$

Find a common denominator

$$v^2 = \frac{\mu}{a}\left(\frac{2+2e\cos\nu-(1-e^2)}{(1-e^2)}\right)$$

Simplify

$$v^2 = \frac{\mu}{a}\left(\frac{1+2e\cos\nu+e^2}{(1-e^2)}\right)$$

Factor the denominator and take square root

$$\boxed{v = \sqrt{\frac{\mu}{a}\left(\frac{1+2e\cos\nu+e^2}{(1-e)(1+e)}\right)}}$$

4. Show that

$$v_{perigee} = \sqrt{\frac{\mu}{a}\left(\frac{1+e}{1-e}\right)}$$

$$v_{apogee} = \sqrt{\frac{\mu}{a}\left(\frac{1-e}{1=e}\right)}$$

At perigee, υ=0°, so cosυ=1, and

$$v_{perigee} = \sqrt{\frac{\mu}{a}\left(\frac{1+2e+e^2}{(1-e)(1+e)}\right)} = \sqrt{\frac{\mu}{a}\left(\frac{(1+e)^2}{(1-e)(1+e)}\right)} = \sqrt{\frac{\mu}{a}\left(\frac{(1+e)}{(1-e)}\right)}$$

At apogee, υ=180°, so cosυ= -1, and

$$v_{apogee} = \sqrt{\frac{\mu}{a}\left(\frac{1-2e+e^2}{(1-e)(1+e)}\right)} = \sqrt{\frac{\mu}{a}\left(\frac{(1-e)^2}{(1-e)(1+e)}\right)} = \sqrt{\frac{\mu}{a}\left(\frac{(1-e)}{(1+e)}\right)}$$

QED

MATLAB Solutions for Homework 2

Problem 1:

MATLAB code for this problem was adapted and extended from the code used for Problem 1 of the Week 1 homework. Additional code was added to achieve the HW2 results.

Results

```
Week 2 Homework Solutions

VECTOR 1
------------------------
r:      326151.080726   6077471.251787   2944583.918767  meters
rd:      -7455.178720      -482.482572      1910.883434 m/sec

h:   1.3034049558e+10 -2.2575636068e+10 4.5151272135e+10 m^2/sec

|h|: 5.2136198243e+10 m^2/sec

p:   6819317.99903984 m

E:   -2.9222906294e+07 m^2/s^2

a:   6819999.99902560 m

TP:     5605.15391191 sec

B:   2.1265080535e+12 3.2207421496e+12 9.9650107635e+11

|B|: 3.9860043767e+12

e:       0.0099999999

nu:      0.5337080028 rad
nu:     30.5792160524 deg
Dot product dot(r, rd) > 0.  Sign is positive.  Quadrant 1 or 2
E0:      0.5286424011 rad
E0:     30.2889784571 deg
M:       0.5235987859 rad
n:       0.0011209657 rad/sec
dt:    467.0961685125 sec
r65: 6790619.6008190252 m
sin(E65):     0.9024485581 rad
cos(E65):     0.4307976322 rad
E65:     1.1254198828 rad
E65:    64.4818094624 deg
dt:    528.8267149214 sec


Kepler Equation Solutions

Echeck:      1.1254198828 rad
Echeck:     64.4818094624 deg
```

```
nuCheck:      1.1344640138 rad
nuCheck:     65.0000000000 deg

E2700:      3.5462689773 rad
E2700:    203.1862454165 deg
Perigee crossings after 2700 seconds: 0
nu2700:      3.5423496855 rad
nu2700:    202.9616865415 deg

E2TP:     13.0950130155 rad
E2TP    750.2889784571 deg
Perigee crossings after 1.121031e+04 seconds: 2

In range [0, 2*pi]
E2TP:      0.5286424011 rad
E2TP:     30.2889784571 deg
nu2TP:      0.5337080028 rad
nu2TP:     30.5792160524 deg

E15000:     17.3280965310 rad
E15000:    992.8267982226 deg
Perigee crossings after 15000 seconds: 2

In range [0, 2*pi]
E15000:      4.7617259167 rad
E15000:    272.8267982226 deg
nu15000:      4.7517354548 rad
nu15000:    272.2543869211 deg


Week 2 Problem 2 Homework Solutions



VECTOR 2
------------------------
R0:     572461.711228  -1015437.194396   7707337.871302  meters
R0d:     -6195.262945     -3575.889650        -5.423283 m/sec

h:   2.7566096726e+10 -4.7745880097e+10 -8.3379603316e+09 m^2/sec

|h|: 5.5759127840e+10 m^2/sec

p:   7799992.201199780 m

E:   -2.5551310368e+07 m^2/s^2

a:   7800000.00120126 m

TP:     6855.71704370 sec

B:   2.8359372629e+11 1.1949255982e+11 2.5333469724e+11

|B|: 3.9860047952e+11

e:       0.0010000001
```

```
nu:      0.8741978248 rad
nu:     50.0878458185 deg

Dot product dot(r, rd) > 0.  Sign is positive.  Quadrant 1 or 2
R0p:   5001362.438709   5978984.522949         0.000000  meters
V0p:     -5483.194150      4593.787225         0.000000  m/sec

f:         0.8386899812 deg
g:       593.8138283683 deg
gdot:      0.8387740125 deg
fdot1:    -0.0004993630 deg
fdot2:    -0.0004993630 deg

Rp:     938596.059564   7742368.795881         0.000000  meters
Vp:      -7096.655979       867.465852         0.000000  m/sec

Test
Rp:     938596.059564   7742368.795881         0.000000  meters
Vp:      -7096.655979       867.465852         0.000000  m/sec
```

MATLAB Code for Problem 1

```
fid = fopen('c:\temp\HW2results 2024.txt','wt');

fprintf(fid,'Week 2 Homework Solutions\n\n');

mu = 3.986004418e14;

fprintf(fid, 'VECTOR 1\n------------------------\n');
r = [326151.080726; 6077471.251787; 2944583.918767];
rd = [-7455.178720; -482.482572; 1910.883434];

fprintf(fid,'r:   %16.6f %16.6f %16.6f  meters\n', r(1), r(2), r(3));
fprintf(fid,'rd:  %16.6f %16.6f %16.6f m/sec\n', rd(1), rd(2), rd(3));
fprintf(fid,'\n');

rnorm = norm(r);

h=cross(r,rd);
fprintf(fid,'h:   %16.10e %16.10e %16.10e m^2/sec\n', h(1), h(2), h(3));
fprintf(fid,'\n');

hnorm = norm(h);
fprintf(fid,'|h|: %16.10e m^2/sec\n', hnorm);
fprintf(fid,'\n');

p = hnorm^2/mu;
fprintf(fid,'p:   %16.8f m\n', p);
fprintf(fid,'\n');

energy = dot(rd,rd)/2 - mu/rnorm;
fprintf(fid,'E:   %16.10e m^2/s^2\n', energy);
fprintf(fid,'\n');

a = -mu/(2*energy);
fprintf(fid,'a:   %16.8f m\n', a);
fprintf(fid,'\n');

TP=2*pi*sqrt(a^3/mu);
fprintf(fid,'TP:  %16.8f sec\n', TP);
fprintf(fid,'\n');

B = cross(rd,h)-(mu/rnorm)*r;
fprintf(fid,'B:   %16.10e %16.10e %16.10e \n', B(1), B(2), B(3));
fprintf(fid,'\n');
Bnorm = norm(B);

fprintf(fid,'|B|: %16.10e \n', Bnorm);
fprintf(fid,'\n');

e = Bnorm/mu;
fprintf(fid,'e:   %16.10f \n', e);
fprintf(fid,'\n');

cosnu = dot(r,B)/(rnorm*Bnorm);
nu = acos(cosnu);
fprintf(fid,'nu:  %16.10f rad\n', nu);
nud = acosd(cosnu);
fprintf(fid,'nu:  %16.10f deg\n', nud);

%Check sign
sign = dot(r, rd);
if(sign<0)
    fprintf(fid,'Dot product dot(r, rd) < 0.  Sign is negative.  Quadrant 3 or 4\n');
else
    fprintf(fid,'Dot product dot(r, rd) > 0.  Sign is positive.  Quadrant 1 or 2\n');
end

%Adjust for quadrant if sign is negative
if sign < 0.
```

```
    nu = 2*pi - nu;
    fprintf(fid,'nu:      %16.8f (adjusted) radians\n', nu);
    nud = 360. - nud;
    fprintf(fid,'nud:     %16.8f (adjusted) degrees\n', nud);
end

% Begin computing initial eccentric anomaly
sinE = sin(nu)*sqrt(1-e^2)/(1 + e*cos(nu));
cosE = (e+cos(nu))/(1+e*cos(nu));

% Use ATAN2 to get E0 in range of [-pi,pi]
E0 = atan2(sinE, cosE);

% Logic to put it in range [0,2pi]
if(E0<0)
    E0 = E0 + 2*pi;
end

fprintf(fid,'E0:  %16.10f rad\n', E0);
fprintf(fid,'E0:  %16.10f deg\n', E0*180/pi);

% Initial mean anomaly
M0 = E0 - e*sin(E0);
fprintf(fid,'M:   %16.10f rad\n', M0);

% Mean motion
n = sqrt(mu/a^3);
fprintf(fid,'n:   %16.10f rad/sec\n', n);

% Time of flight from perigee to initial NU
dt_per = M0/n;
fprintf(fid,'dt:  %16.10f sec\n', dt_per);

% Now compute radius, eccentric anomaly, and other
% parameters for nu = 65 degrees
nu65 = 65*pi/180.;

% Need to compute radius of orbit at nu=65 deg
r65 = a*(1-e^2)/(1+e*cos(nu65));
fprintf(fid,'r65: %16.10f m\n', r65);

% Compute eccentric anomaly for nu=65
% sinE65 = r65*sin(nu65)/(a*sqrt(1-e^2));
sinE65 = sin(nu65)*sqrt(1-e^2)/(1 + e*cos(nu65));
cosE65 = (e+cos(nu65))/(1+e*cos(nu65));

% Use ATAN2 to get E65 in range of [-pi,pi]
E65 = atan2(sinE65, cosE65);

% Logic to put it in range [0,2pi]
if(E65<0)
    E65 = E65 + 2*pi;
end
fprintf(fid,'sin(E65): %16.10f rad\n', sinE65);
fprintf(fid,'cos(E65): %16.10f rad\n', cosE65);

fprintf(fid,'E65: %16.10f rad\n', E65);
fprintf(fid,'E65: %16.10f deg\n', E65*180/pi);

% compute time of flight from initial nu, to nu=65 deg
% Here we assume no perigee crossing between the two points
dt65 = 1/n*(E65-e*sin(E65))-1/n*(E0-e*sin(E0));
fprintf(fid,'dt:  %16.10f sec\n', dt65);

fprintf(fid,'\n\nKepler Equation Solutions\n\n');

% Solution using time of flight computed above
dt = dt65;
Echeck = Kepler_Equation( mu, a, e, M0, dt);
fprintf(fid,'Echeck:  %16.10f rad\n', Echeck);
fprintf(fid,'Echeck:  %16.10f deg\n', Echeck*180/pi);
```

```
nuCheck = nu_from_E(Echeck, e);
fprintf(fid,'nuCheck:  %16.10f rad\n', nuCheck);
fprintf(fid,'nuCheck:  %16.10f deg\n', nuCheck*180/pi);

% Solution using time of flight = 2700 seconds
dt = 2700;
E2700 = Kepler_Equation( mu, a, e, M0, dt);
fprintf(fid,'\nE2700:  %16.10f rad\n', E2700);
fprintf(fid,'E2700:  %16.10f deg\n', E2700*180/pi);

numPerigeeCrossings = floor((E2700 - E0) / (2*pi));
fprintf(fid,'Perigee crossings after %d seconds: %d\n', dt, numPerigeeCrossings);

nu2700 = nu_from_E(E2700, e);
fprintf(fid,'nu2700:  %16.10f rad\n', nu2700);
fprintf(fid,'nu2700:  %16.10f deg\n', nu2700*180/pi);

% Solution using time of flight = exactly two orbit periods
dt = 2 * TP;
E2TP = Kepler_Equation( mu, a, e, M0, dt);
fprintf(fid,'\nE2TP:  %16.10f rad\n', E2TP);
fprintf(fid,'E2TP  %16.10f deg\n', E2TP*180/pi);

numPerigeeCrossings = floor((E2TP - E0) / (2*pi));
fprintf(fid,'Perigee crossings after %d seconds: %d\n', dt, numPerigeeCrossings);

fprintf(fid,'\nIn range [0, 2*pi]\n');
E2TP = mod(E2TP, 2*pi);
fprintf(fid,'E2TP:  %16.10f rad\n', E2TP);
fprintf(fid,'E2TP:  %16.10f deg\n', E2TP*180/pi);

nu2TP = nu_from_E(E2TP, e);
fprintf(fid,'nu2TP:  %16.10f rad\n', nu2TP);
fprintf(fid,'nu2TP:  %16.10f deg\n', nu2TP*180/pi);

% Solution using time of flight = 1500 seconds
dt = 15000;
E15000 = Kepler_Equation( mu, a, e, M0, dt);
fprintf(fid,'\nE15000:  %16.10f rad\n', E15000);
fprintf(fid,'E15000:  %16.10f deg\n', E15000*180/pi);

numPerigeeCrossings = floor((E15000 - E0) / (2*pi));
fprintf(fid,'Perigee crossings after %d seconds: %d\n', dt, numPerigeeCrossings);

fprintf(fid,'\nIn range [0, 2*pi]\n');
E15000 = mod(E15000, 2*pi);
fprintf(fid,'E15000:  %16.10f rad\n', E15000);
fprintf(fid,'E15000:  %16.10f deg\n', E15000*180/pi);

nu15000 = nu_from_E(E15000, e);
fprintf(fid,'nu15000:  %16.10f rad\n', nu15000);
fprintf(fid,'nu15000:  %16.10f deg\n', nu15000*180/pi);
```

Solution of Kepler Equation

```
function [ E1 ] = Kepler_Equation( mu, a, e, M0, dt)
% Solves Kepler's equation to find E1 given an initial MA and dt

    n = sqrt(mu/a^3); % Mean motion
    tolerance = 1e-10;
    update = 100000; % Initialize to a big number

    Ek = M0; % As a first estimate, set Ek to input mean anomaly

    iteration = 0;

% The logic uses the absolute value of the update, which could be positive
% or negative.
    while (abs(update)>tolerance && (iteration<=10));

% Newtonian iteration has the following form:
%
%       Ek+1 = Ek - f(E)/f'(E)
%
% Function and derivative are as shown (see, e.g., Vallado sect 2.2.5)
        f=n*dt+M0-(Ek-e*sin(Ek));
        fp= -(1-e*cos(Ek));

% We compute update as a separate variable for use in the "While" logic
        update = f/fp;

% Compute updated estimate
        Ekp1=Ek-update;
        iteration = iteration+1;

% Set up Ek for the next iteration
        Ek=Ekp1;
    end

% Set the return value to the converged
    E1=Ekp1;
end
```

Compute True Anomaly from Eccentric Anomaly

```
function [ nu ] = nu_from_E( E, e)
% Compute nu given E and e
% E - eccentric anomaly in RADIANS
    cosnu = (cos(E)-e)/(1-e*cos(E));
    sinnu = sin(E)*sqrt(1-e^2)/(1-e*cos(E));

    nu = atan2(sinnu, cosnu);
    if(nu<0)
        nu = nu + 2*pi;
    end

end
```

Problem 2:

MATLAB code for this problem was adapted and extended from the code used for Problem 2 of the Week 1 homework. Additional code was added to achieve the HW2 results.

Results

```
Week 2 Homework Solutions

VECTOR 2
------------------------
R0:     572461.711228  -1015437.194396   7707337.871302  meters
R0d:     -6195.262945     -3575.889650        -5.423283 m/sec

Keplerian Elements for Vector 2
a:     7800000.00120126 m
e:           0.00100000
inc:        98.60000000 deg
raan:       30.00000000 deg
wp:         40.00000696 deg
nu:         50.08784582 deg

h:   2.7566096726e+10 -4.7745880097e+10 -8.3379603316e+09 m^2/sec

|h|: 5.5759127840e+10 m^2/sec

p:   7799992.201199780 m

E:   -2.5551310368e+07 m^2/s^2

a:   7800000.00120126 m

TP:     6855.71704370 sec

B:   2.8359372629e+11 1.1949255982e+11 2.5333469724e+11

|B|: 3.9860047952e+11

e:       0.0010000001

nu:      0.8741978248 rad
nu:     50.0878458185 deg

Dot product dot(r, rd) > 0.  Sign is positive.  Quadrant 1 or 2
R0p:   5001362.438709   5978984.522949         0.000000  meters
V0p:     -5483.194150      4593.787225         0.000000  m/sec

f:         0.8386899812 deg
g:       593.8138283683 deg
gdot:      0.8387740125 deg
fdot1:    -0.0004993630 deg
fdot2:    -0.0004993630 deg

Rp:     938596.059564   7742368.795881         0.000000  meters
Vp:      -7096.655979       867.465852         0.000000  m/sec

Test
Rp:     938596.059564   7742368.795881         0.000000  meters
Vp:      -7096.655979       867.465852         0.000000  m/sec
```

MATLAB Code for Problem 2

```
fprintf(fid,'\n\nWeek 2 Homework Solutions\n\n');

mu = 3.986004418e14;
fprintf(fid, '\n\nVECTOR 2\n------------------------\n');
R0=[572461.711228; -1015437.194396; 7707337.871302];
R0d=[-6195.262945; -3575.889650; -5.423283];

fprintf(fid,'R0:  %16.6f %16.6f %16.6f  meters\n', R0(1), R0(2), R0(3));
fprintf(fid,'R0d: %16.6f %16.6f %16.6f m/sec\n', R0d(1), R0d(2), R0d(3));
fprintf(fid,'\n');

% Compute the Keplerian elements
[a, e, inc, raan, wp, nu] = KeplerianElements(mu, R0, R0d);

WriteKeplerianElements(fid, 2, a, e, inc, raan, wp, nu);
r0 = norm(R0);

h=cross(R0,R0d);
fprintf(fid,'\n');
fprintf(fid,'h:   %16.10e %16.10e %16.10e m^2/sec\n', h(1), h(2), h(3));
fprintf(fid,'\n');

hnorm = norm(h);
fprintf(fid,'|h|: %16.10e m^2/sec\n', hnorm);
fprintf(fid,'\n');

p = hnorm^2/mu;
fprintf(fid,'p:   %16.9f m\n', p);
fprintf(fid,'\n');

energy = dot(R0d,R0d)/2 - mu/r0;
fprintf(fid,'E:   %16.10e m^2/s^2\n', energy);
fprintf(fid,'\n');

a = -mu/(2*energy);
fprintf(fid,'a:   %16.8f m\n', a);
fprintf(fid,'\n');

TP=2*pi*sqrt(a^3/mu);
fprintf(fid,'TP:  %16.8f sec\n', TP);
fprintf(fid,'\n');

B = cross(R0d,h)-(mu/r0)*R0;
fprintf(fid,'B:   %16.10e %16.10e %16.10e \n', B(1), B(2), B(3));
fprintf(fid,'\n');
Bnorm = norm(B);

fprintf(fid,'|B|: %16.10e \n', Bnorm);
fprintf(fid,'\n');

e = Bnorm/mu;
fprintf(fid,'e:   %16.10f \n', e);
fprintf(fid,'\n');

cosnu = dot(R0,B)/(r0*Bnorm);
nu = acos(cosnu);
fprintf(fid,'nu:  %16.10f rad\n', nu);
nud = acosd(cosnu);
fprintf(fid,'nu:  %16.10f deg\n', nud);
fprintf(fid,'\n');

%Check sign
sign = dot(R0, R0d);
if(sign<0)
    fprintf(fid,'Dot product dot(r, rd) < 0.  Sign is negative.  Quadrant 3 or 4\n');
else
```

```
    fprintf(fid,'Dot product dot(r, rd) > 0.  Sign is positive.  Quadrant 1 or 2\n');
end

%Adjust for quadrant if sign is negative
if sign < 0.
    nu = 2*pi - nu;
    fprintf(fid,'nu:      %16.8f (adjusted) radians\n', nu);
    nud = 360. - nud;
    fprintf(fid,'nud:     %16.8f (adjusted) degrees\n', nud);
end

% Compute perifocal vector corresponding to the input vector
x = r0*cos(nu);
y = r0*sin(nu);
xdot = -sqrt(mu/p)*sin(nu);
ydot = sqrt(mu/p)*(e+cos(nu));

% form into vectors r0 and v0
R0p=[x; y; 0.];
V0p=[xdot; ydot; 0.];
fprintf(fid,'R0p: %16.6f %16.6f %16.6f  meters\n', R0p(1), R0p(2), R0p(3));
fprintf(fid,'V0p: %16.6f %16.6f %16.6f  m/sec\n', V0p(1), V0p(2), V0p(3));
fprintf(fid,'\n');

% Find f, g, fdot, gdot for dnu=33 deg
dnu = 33*pi/180.;

% To find f, we need the radius at nu+33 deg
nuplus33=nu+dnu;
r = p/(1+e*cos(nuplus33));
f = 1-(r/p)*(1-cos(dnu));
g = (r*r0)/sqrt(mu*p)*sin(dnu);
gdot = 1-(r0/p)*(1-cos(dnu));

% compute fdot using both available methods
fdot1 = (f*gdot-1)/g;
fdot2 = sqrt(mu/p)*tan(dnu/2)*((1-cos(dnu))/p - 1/r - 1/r0);
fprintf(fid,'f:     %16.10f deg\n', f);
fprintf(fid,'g:     %16.10f deg\n', g);
fprintf(fid,'gdot:  %16.10f deg\n', gdot);
fprintf(fid,'fdot1: %16.10f deg\n', fdot1);
fprintf(fid,'fdot2: %16.10f deg\n', fdot2);
fprintf(fid,'\n');

% Now use the f and g functions to compute R and V at dnu=33 deg
% Remember to use the perifocal vectors R0p and V0p
Rp = f*R0p + g*V0p;
Vp = fdot1*R0p + gdot*V0p;
fprintf(fid,'Rp:  %16.6f %16.6f %16.6f  meters\n', Rp(1), Rp(2), Rp(3));
fprintf(fid,'Vp:  %16.6f %16.6f %16.6f  m/sec\n', Vp(1), Vp(2), Vp(3));
fprintf(fid,'\n');

% Check answers using the x, y, xdot, ydot equations
x = r*cos(nuplus33);
y = r*sin(nuplus33);
xdot = -sqrt(mu/p)*sin(nuplus33);
ydot = sqrt(mu/p)*(e+cos(nuplus33));

% form into vectors r0 and v0
Rp=[x; y; 0.];
Vp=[xdot; ydot; 0.];
fprintf(fid,'Test\n');
fprintf(fid,'Rp:  %16.6f %16.6f %16.6f  meters\n', Rp(1), Rp(2), Rp(3));
fprintf(fid,'Vp:  %16.6f %16.6f %16.6f  m/sec\n', Vp(1), Vp(2), Vp(3));

fclose(fid);
```