Photo 2 Homework 2, due Friday 19 Feb

1. Write two matlab functions, one to convert a state vector (ECI) into kepler elements, and another to convert kepler elements into the corresponding state vector (ECI). Use the algorithms given in class. Demonstrate functionality by converting the given state vector to kepler elements and back (you should get what you started with) (5120116.9051000001, 371080.7105000000, 4579618.9369000001, 4891.8051000000, -1681.3341000000, -5586.9748000000) Units are meters and meters per second, data from satellite EROS-A. Note: you will use these functions for subsequent tasks.

2. The following state vector is given for Worldview-1 in ECF at the given time in UTC. Use the supplied matlab functions to convert this to an ECI state vector. (236729.708388, -5193196.562968, 4488000.563859, -1461.122913, -4984.744868, -5674.861645), 2008-07-01 (1-July-2008) 16:44:21.171960(Z)

3. Starting at the point given in problem 2, plot the ground track of the orbit trajectory for two days, at intervals of one minute, overlaid on the supplied world coastline data set. Plot coastline in blue, day 1 in red, day 2 in green.

```
% wv1_orb.m  12-feb-10
% code template for initial part of plotting satellite
% ground track for worldview-1
% if you want details of the ecf <=> eci transformation,
% see references: tapley,schutz,born or montenbruck

% define state vector (ecf) for worldview ephemeris point #1


% date/time (year/month/day time(seconds of day)) given for WV1 eph #1
% time: hr*3600 + min*60 + s
Y=
M=
D=
UTC=

% get data from www.iers.org
% from bulletin B, time and polar motion
UT1_m_UTC=
xp=
yp=
% from bulletin C,cumulative leap seconds
UTC_m_TAI=

% fixed constant time shift
TT_m_TAI=32.184; % (s)

% derived
TT_m_UTC=TT_m_TAI - UTC_m_TAI; % (s)
TT=UTC + TT_m_UTC; % (s)


% compute julian day from date
tsecperday=24*60*60;
Df=D+TT/tsecperday;
JD=jd(Y,M,Df);
% julian century
T=(JD-2451545)/36525;
% compute precession and nutation (1980 model)
P=precession(T);
[N,ep,delpsi]=nutation(T);
JDu=jd(Y,M,D);
Tu=(JDu-2451545)/36525;
UT1=UTC + UT1_m_UTC;
% UTC, UT1 in time seconds
% compute greenwich apparent sidereal time
GAST=compGAST(Tu,ep,delpsi,UT1);


% compute THETA and PI
THETA=
PI=
% compute R_FI for Xecf = R_FI * Xeci

% convert state vector Xecf to state vector Xeci

% get kepler elements (your function)

% loop: advance time and satellite position and earth position
% save satellite ground position in "plot vector", then
% filter and plot - see posted detailed algorithm
```

Algorithm for plotting satellite ground track for WV-1 (see wv1_orb.m for preliminaries)

```
Get initial point in ECF
Compute GAST and P,N,θ,π
Convert point to ECI
Convert to Kepler: a,e,Ω,ω,i,f
Get auxilliary orbit info: τ, $t_p$, $\Delta_{PS}$, n, t (sec of day)
Current GAST, (radians): g
Loop for 2 days at 1 minute interval
t = t + increment
if $(t – t_p) > \tau$   then
     $t_p = t_p + \tau$
Compute M, E, f (new one)
Kepler to state vector
g = g + increment * (366.2524/365.2524)
$\Theta = R_3(g)$  (new)
P, N, π stay the same
New R
Convert ECI to ECF, position only
Scale ECF vector to nominal earth radius
[ϕ, λ, h]=xyz2geo(X,Y,Z)
Convert λ and ϕ to degrees
Save plot points in arrays px, py (units are decimal degrees)
When finished filling in px,py,
Loop through all px, py starting at #2
Only plot line segment if
     $| py_i | < 85$  (it’s crowded at poles)
     $py_i < py_{i-1}$   (plot only descending pass)
     $| px_i – px_{i-1} | < 180$  (avoid wrap around)
If day one
Plot([px(i-1) px(i)],[py(i-1) py(i)],’r-’);
If day two
Plot([px(i-1) px(i)],[py(i-1) py(i)],’g-’);
```

Get useful matlab and datafiles from orbit.zip

```
Load coast
Plot(long,lat,’b-’);
Axis equal
```

(Note: used “f” instead of “v” for true anomaly for this slide)