	Computed from single ECF state vector	Digital Globe nominal published values	NORAD TLE derived (different date)
Inclination	97.2 deg	97.2 deg	97.2 deg
Altitude (a-Re)	456 km	450 km	457 km
eccentricity	0.00136		0.00008
Period	93.4 min	93.5 min	93.5 min

```
qb_eci_ecf
P =
         0.999998639336414      -0.00151297894345682     -0.000657434436405391
       0.00151297894345302          0.99999885544658    -4.97348351562241e-007
      0.000657434436414142    -4.97336784143812e-007         0.999999783889834
N =
         0.999999999993421    -3.32810870732286e-006    -1.44285100297868e-006
     3.32804117583298e-006         0.999999998899264    -4.68016782524949e-005
     1.44300676246339e-006     4.68016734503363e-005         0.999999998903761
THETA =
       0.00941013070686709        -0.999955723739846                         0
         0.999955723739846       0.00941013070686709                         0
                         0                         0                         1
PI =
         0.999999999999993     1.45763680268058e-013     1.18197575454415e-007
                         0          0.99999999999924    -1.23322056063801e-006
    -1.18197575454505e-007     1.23322056063801e-006         0.999999999999233
R_FI =
       0.00789390885014488        -0.999968841766687     4.12171863114538e-005
         0.999968624690182       0.00789387991061955    -0.000660526443622518
      0.000660180499266482     4.64300286485467e-005         0.999999781002957
R_FI_T =
       0.00789390885014488         0.999968624690182      0.000660180499266482
        -0.999968841766687       0.00789387991061955     4.64300286485467e-005
     4.12171863114538e-005    -0.000660526443622518         0.999999781002957
XX_eci =
         -5168737.22542626
         -421617.792531673
          4430412.86504082
XX_eci_dot =
         -4992.21801898959
          861.996867566927
         -5726.56854169279
kp =
       -0.0271524135346729
          1.69692600641768
          1.09886593882228
          1.32886997953879
          6822955.02822704
       0.00136258389599335
          5608.79687213329
          1.32622482173429
          1.32754729190325
          978.756428515719
          1183.87812363087
          59857.6276723691
diary off
```

```
% qb_eci_ecf.m  6-feb-08
% computation of rotation matrix to transform
% ECf to ECI, convert both position and velocity
% follow example of exercises 5.1,5.2 montenbruck, p.190
% for quickbird lafayette scene, eph. point #5
% start epoch: 2006 10 07 16:57:21.425796Z + 4*0.020 (=0.08)
% October 7, 2006 16:57:21 GMT, (11:57:21 EST)
% actual epoch for transformation = 16:57:21.505796

% XYZ,XvYvZv, (m) (m/s)
X_ecf=  380985.72429075348;
Y_ecf=-5174829.659770391;
Z_ecf= 4426980.0195452599;
Xv_ecf=-1278.9683323438;
Yv_ecf=-5009.256219147137;
Zv_ecf=-5729.823019695019;
% Bulletin B from www.iers.org
% UT1-UTC=0.137195 (s)
% xp=0.02438 (")
% yp=0.25437 (")
% Bulletin C from www.iers.org
% UTC-TAI=-33.0 (s)
% Fixed constant
% TT-TAI=32.184 (s)

% given
Y=2006;
M=10;
D=7;
UT1_m_UTC=0.137195;
xp=0.02438; % (")
yp=0.25437; % (")
UTC_m_TAI=-33; % (s)
UTC=16*3600 + 57*60 + 21.505796; % (s)
TT_m_TAI=32.184; % (s)

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

degrad=180/pi;
secrad=degrad*3600;
```

```
xpR=xp/secrad;
ypR=yp/secrad;

tsecperday=24*60*60;
Df=D+TT/tsecperday;
JD=jd(Y,M,Df);
T=(JD-2451545)/36525;
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
GAST=compGAST(Tu,ep,delpsi,UT1);
GAST
pause

THETA=m3(GAST);
% following two expressions are equivalent
PI=m2(-xpR)*m1(-ypR);
%PI=[1 0 xpR; 0 1 -ypR;-xpR ypR 1];
R_FI=PI * THETA * N * P;
P
N
THETA
PI
R_FI
% this one multiplies Xecf to produce Xeci
R_FI_T=R_FI'
XX_ecf=[X_ecf;Y_ecf;Z_ecf];
XX_eci=R_FI_T*XX_ecf

% ok, now let's get the velocity
XX_ecf_dot=[Xv_ecf;Yv_ecf;Zv_ecf];
om=7.292115856e-05;
pre=[0 -1 0;1 0 0;0 0 0];
R_FI_T_dot=P' * N' * pre * THETA' * PI' * om;
XX_eci_dot=R_FI_T_dot*XX_ecf + R_FI_T*XX_ecf_dot

% ok now get kepler elements

kp=stkp6(XX_eci,XX_eci_dot,UTC)
```

```
% precession.m  1-feb-08
% compute precession matrix using IAU 1980 model
function result=precession(T)
degrad=180/pi;
secrad=degrad*3600;
zetas=2306.2181*T + 0.30188*T^2 + 0.017998*T^3;
zs=2306.2181*T + 1.09468*T^2 + 0.018203*T^3;
thetas=2004.3109*T - 0.42665*T^2 - 0.041833*T^3;
zeta=zetas/secrad;
z=zs/secrad;
theta=thetas/secrad;
result=m3(-z)*m2(theta)*m3(-zeta);
```

```
% nutation.m  2-feb-08
% compute nutation matrix by IAU 1980 model
% coefficients for 106 term expression for delta-psi
% and delta-eps comes from montenbruck
% columns in the table are
% pl plp pF pD pOm dP0 dP1 de0 de1 indx
function [N,ep,delpsi]=nutation(T)
[pl,plp,pF,pD,pOm,dP0,dP1,de0,de1,indx]=textread('iau1980n.txt','%f%f%f%↙
f%f%f%f%f%f%f');
dP1=dP1/10;
de1=de1/10;
degrad=180/pi;
secrad=degrad*3600;
% l,lp,F,D,Om all in radians
c0=(134+57/60+46.733/3600)/degrad;
c1=(477198+52/60+2.633/3600)/degrad;
c2=(31.310/3600)/degrad;
c3=(0.064/3600)/degrad;
l=c0 + c1*T + c2*T^2 + c3*T^3;
c0=(357+31/60+39.804/3600)/degrad;
c1=(35999+3/60+1.224/3600)/degrad;
c2=(-0.577/3600)/degrad;
c3=(-0.012/3600)/degrad;
lp=c0 + c1*T + c2*T^2 + c3*T^3;
c0=(93+16/60+18.877/3600)/degrad;
c1=(483202+1/60+3.137/3600)/degrad;
c2=(-13.257/3600)/degrad;
c3=(0.011/3600)/degrad;
F=c0 + c1*T + c2*T^2 + c3*T^3;
c0=(297+51/60+1.307/3600)/degrad;
c1=(445267+6/60+41.328/3600)/degrad;
c2=(-6.891/3600)/degrad;
c3=(0.019/3600)/degrad;
D=c0 + c1*T + c2*T^2 + c3*T^3;
c0=(125+2/60+40.280/3600)/degrad;
c1=(-(1934+8/60+10.539/3600))/degrad;
c2=(7.455/3600)/degrad;
c3=(0.008/3600)/degrad;
Om=c0 + c1*T + c2*T^2 + c3*T^3;
delpsi=0;
deleps=0;
for i=1:106
```

```
  phi=pl(i)*l + plp(i)*lp + pF(i)*F + pD(i)*D + pOm(i)*Om;
  % phi in radians
  dP=dP0(i) + dP1(i)*T;
  delpsi=delpsi + dP*sin(phi);
  de=de0(i) + de1(i)*T;
  deleps=deleps + de*cos(phi);
end
% put delpsi and deleps into radians
delpsi=delpsi/10000;
delpsi=delpsi/secrad;
deleps=deleps/10000;
deleps=deleps/secrad;
c0=23.43929111/degrad;
c1=(-46.8150/3600)/degrad;
c2=(-0.00059/3600)/degrad;
c3=(0.001813/3600)/degrad;
ep=c0 + c1*T + c2*T^2 + c3*T^3;
% eps in radians
N=m1(-ep-deleps)*m3(-delpsi)*m1(ep);
```

0	0	0	0	1	-171996	-1742	92025	89	1
0	0	0	0	2	2062	2	-895	5	2
-2	0	2	0	1	46	0	-24	0	3
2	0	-2	0	0	11	0	0	0	4
-2	0	2	0	2	-3	0	1	0	5
1	-1	0	-1	0	-3	0	0	0	6
0	-2	2	-2	1	-2	0	1	0	7
2	0	-2	0	1	1	0	0	0	8
0	0	2	-2	2	-13187	-16	5736	-31	9
0	1	0	0	0	1426	-34	54	-1	10
0	1	2	-2	2	-517	12	224	-6	11
0	-1	2	-2	2	217	-5	-95	3	12
0	0	2	-2	1	129	1	-70	0	13
2	0	0	-2	0	48	0	1	0	14
0	0	2	-2	0	-22	0	0	0	15
0	2	0	0	0	17	-1	0	0	16
0	1	0	0	1	-15	0	9	0	17
0	2	2	-2	2	-16	1	7	0	18
0	-1	0	0	1	-12	0	6	0	19
-2	0	0	2	1	-6	0	3	0	20
0	-1	2	-2	1	-5	0	3	0	21
2	0	0	2	1	4	0	-2	0	22
0	1	2	-2	1	4	0	-2	0	23
1	0	0	-1	0	-4	0	0	0	24
2	1	0	-2	0	1	0	0	0	25
0	0	-2	2	1	1	0	0	0	26
0	1	-2	2	0	-1	0	0	0	27
0	1	0	0	2	1	0	0	0	28
-1	0	0	1	1	1	0	0	0	29
0	1	2	-2	0	-1	0	0	0	30
0	0	2	0	2	-2274	-2	977	-5	31
1	0	0	0	0	712	1	-7	0	32
0	0	2	0	1	-386	-4	200	0	33
1	0	2	0	2	-301	0	129	-1	34
1	0	0	-2	0	-158	0	-1	0	35
-1	0	2	0	2	123	0	-53	0	36
0	0	0	2	0	63	0	-2	0	37
1	0	0	0	1	63	1	-33	0	38
-1	0	0	0	1	-58	-1	32	0	39
-1	0	2	2	2	-59	0	26	0	40
1	0	2	0	1	-51	0	27	0	41
0	0	2	2	2	-38	0	16	0	42
2	0	0	0	0	29	0	-1	0	43
1	0	2	-2	2	29	0	-12	0	44
2	0	2	0	2	-31	0	13	0	45
0	0	2	0	0	26	0	-1	0	46
-1	0	2	0	1	21	0	-10	0	47
-1	0	0	2	1	16	0	-8	0	48
1	0	0	-2	1	-13	0	7	0	49
-1	0	2	2	1	-10	0	5	0	50
1	1	0	-2	0	-7	0	0	0	51
0	1	2	0	2	7	0	-3	0	52
0	-1	2	0	2	-7	0	3	0	53
1	0	2	2	2	-8	0	3	0	54
1	0	0	2	0	6	0	0	0	55

2	0	2	-2	2	6	0	-3	0	56
0	0	0	2	1	-6	0	3	0	57
0	0	2	2	1	-7	0	3	0	58
1	0	2	-2	1	6	0	-3	0	59
0	0	0	-2	1	-5	0	3	0	60
1	-1	0	0	0	5	0	0	0	61
2	0	2	0	1	-5	0	3	0	62
0	1	0	-2	0	-4	0	0	0	63
1	0	-2	0	0	4	0	0	0	64
0	0	0	1	0	-4	0	0	0	65
1	1	0	0	0	-3	0	0	0	66
1	0	2	0	0	3	0	0	0	67
1	-1	2	0	2	-3	0	1	0	68
-1	-1	2	2	2	-3	0	1	0	69
-2	0	0	0	1	-2	0	1	0	70
3	0	2	0	2	-3	0	1	0	71
0	-1	2	2	2	-3	0	1	0	72
1	1	2	0	2	2	0	-1	0	73
-1	0	2	-2	1	-2	0	1	0	74
2	0	0	0	1	2	0	-1	0	75
1	0	0	0	2	-2	0	1	0	76
3	0	0	0	0	2	0	0	0	77
0	0	2	1	2	2	0	-1	0	78
-1	0	0	0	2	1	0	-1	0	79
1	0	0	-4	0	-1	0	0	0	80
-2	0	2	2	2	1	0	-1	0	81
-1	0	2	4	2	-2	0	1	0	82
2	0	0	-4	0	-1	0	0	0	83
1	1	2	2	2	1	0	-1	0	84
1	0	2	2	1	-1	0	1	0	85
-2	0	2	4	2	-1	0	1	0	86
-1	0	4	0	2	1	0	0	0	87
1	-1	0	-2	0	1	0	0	0	88
2	0	2	-2	1	1	0	-1	0	89
2	0	2	2	2	-1	0	0	0	90
1	0	0	2	1	-1	0	0	0	91
0	0	4	-2	2	1	0	0	0	92
3	0	2	-2	2	1	0	0	0	93
1	0	2	-2	0	-1	0	0	0	94
0	1	2	0	1	1	0	0	0	95
-1	-1	0	2	1	1	0	0	0	96
0	0	-2	0	1	-1	0	0	0	97
0	0	2	-1	2	-1	0	0	0	98
0	1	0	2	0	-1	0	0	0	99
1	0	-2	-2	0	-1	0	0	0	100
0	-1	2	0	1	-1	0	0	0	101
1	1	0	-2	1	-1	0	0	0	102
1	0	-2	2	0	-1	0	0	0	103
2	0	0	2	0	1	0	0	0	104
0	0	2	4	2	-1	0	0	0	105
0	1	0	1	0	1	0	0	0	106

```
% compGAST.m  3-feb-08
% compute GAST greenwich apparent sidereal time
% careful about units
% UT1 in time seconds
% ep and delpsi in radians
function GAST=compGAST(Tu,ep,delpsi,UT1)
tsecrad=(24*60*60)/(2*pi);
c0=24110.54841;
c1=8640184.812866;
c2=0.093104;
c3=-0.0000062;
GMST_0h_UT1=c0 + c1*Tu + c2*Tu^2 + c3*Tu^3;
% GMST time seconds
r=1.002737909350795 + 5.9006e-011*Tu - 5.9e-15*Tu^2;
GMST=GMST_0h_UT1 + r*UT1;
% GMST time seconds
GAST=GMST/tsecrad + delpsi*cos(ep);
% delpsi radians, convert GMST to radians so that
% GAST returned as radians
```

```
% jd.m  1-feb-08
% compute julian day
% days since -4712
function result=jd(Y,M,D)
if(M <= 2)
  Y=Y-1;
  M=M+12;
  end
A=fix(Y/100);
B=2-A+fix(A/4);
term1=fix(365.25*(Y+4716));
term2=fix(30.6001*(M+1));
term3=D + B - 1524.5;
result=term1 + term2 + term3;
```

```
function [stv,f]=kep2stv(kep,t)
%Convert Kepler element to state vector

%↙
=========================================================================↙
=
%Input parameters
% kep: Kepler element, 1x8
%     1) a(km): semi-major axis of the satellite orbit
%     2) e: eccentricity of the orbit
%     3) inc(rad): the orbit inclination
%     4) OMEGA_0(rad): right ascension of ascending node at t=t0
%     5) OMEGA_1(rad/s): time rate of OMEGA, OMEGA=OMEGA_0 + OMEGA_1*(t-↙
t0)
%     6) omega_0(rad): argument of perigee a time = t0
%     7) omega_1(rad/s): time rate of omega, omega=omega_0 + omega_1*(t-↙
t0)
%     8) tp(seconds of day): time at perigee
% t : epoch(sec. of day) of state vector asked to compute
% [Note] unit of tp and t can be seconds from ascending node, just t-tp↙
is
% important in this code

%Output parameter
% stv: state vector, 1x6, (km, km/s)
% f  : true anomaly, (rad)
%↙
=========================================================================↙
=

a=kep(1);
e=kep(2);
i=kep(3);
Om0=kep(4);
Om1=kep(5);
w0=kep(6);
w1=kep(7);
tp=kep(8);
Omega=Om0 + Om1*(t-tp);
w=    w0  + w1 *(t-tp);

GM=3986005*10^8*10^-9; %(km^3/s^2)
```

```
%GM=3.986005e+14;%(m^3/s^2)
n=sqrt(GM/a^3);
Tau=2*pi/n;
M=n*(t-tp);
% iterate to solve kepler eq for E
E0=M;
for j=1:10
  E=M + e*sin(E0);
  E0=E;
  end
num=sqrt(1-e^2)*sin(E);
den=cos(E)-e;
f=atan2(num,den);

r=a*(1-e^2)/(1+e*cos(f));
q=[r*cos(f); r*sin(f); 0];
tmpv=[-sin(f); e+cos(f); 0];
qdot=tmpv * ((n*a)/sqrt(1-e^2));
R3_w=[cos(w) sin(w) 0; -sin(w) cos(w) 0; 0 0 1];
R1_i=[1 0 0;0 cos(i) sin(i);0 -sin(i) cos(i)];
R3_Om=[cos(Omega) sin(Omega) 0; -sin(Omega) cos(Omega) 0; 0 0 1];

Rqx=R3_w*R1_i*R3_Om;
Rxq=Rqx';

X=Rxq*q;
Xdot=Rxq*qdot;
stv=[X;Xdot]';%1x6
```

```
% stkp6.m - 16-feb-06
% function version state vector to kepler element
% this version for eros-a with eci reference frame
% input: position vector X, velocity vector V, and time
% output: kepler elements Omega,i,w,f,a,e
%         also M,E,tau,tp,tau,deltp
% use units: meters

function kp = stkp(XX,VV,t)

% from orbit1.m, ..., stkp5.m
% leick's method x,v -> kepler

u=3.986005e+14; % m^3 s^-2
% XX=[595.4360; -4465.2210; 5614.0670];
% XX=[328.156; -5435.635; 4709.463];
% XX=[408.384884;-5163.749763;4436.609396];
x=sqrt(XX(1)^2 + XX(2)^2 + XX(3)^2);

% V=[-1.0376790; -5.8253970; -4.5130440];
% V=[-1.1972930; -4.8576410; -5.5091720];
% V=[-1.239488853;-5.028787960;-5.720958983];
v=sqrt(VV(1)^2 + VV(2)^2 + VV(3)^2);
% tim=60180;

H=cross(XX,VV);
h=sqrt(H(1)^2 + H(2)^2 + H(3)^2);
Omega=atan2(H(1),-H(2));
i=atan2(sqrt(H(1)^2 + H(2)^2),H(3));
R1=[1 0 0;0 cos(i) sin(i);0 -sin(i) cos(i)];
R3=[cos(Omega) sin(Omega) 0; -sin(Omega) cos(Omega) 0; 0 0 1];
P=R1*R3*XX;
wplusf=atan2(P(2),P(1));
r=x;
a=r/(2 - (r*v*v)/u);
e=sqrt(1 - (h*h)/(u*a) );
cosE=(a-r)/(a*e);
XdotV=XX(1)*VV(1) + XX(2)*VV(2) + XX(3)*VV(3);
sinE=XdotV/(e*sqrt(u*a));
f=atan2(sqrt(1-e^2)*sinE,cosE-e);
E=atan2(sinE,cosE);
M=E - e*sinE;
```

```
w=wplusf - f;

Omega;
i;
a;
e;
E;
M;
f;
w;

Tau=2*pi*sqrt(a^3/u);
Tau;
term1=(Tau/pi)*atan(sqrt(1-e)*tan(w/2)/sqrt(1+e));
term2=(Tau/(2*pi))*e*sqrt(1-e^2)*sin(w)/(1+e*cos(w));
tp=term1-term2;
tp;
%we=0.00007272205;
deltp=M*Tau/(2*pi);
t0=t-deltp;

% put result into returned vector

kp=[Omega;i;w;f;a;e;Tau;M;E;tp;deltp;t0];

% now go back

%q=[r*cos(f); r*sin(f); 0];
%qdot=[-sin(f); e+cos(f); 0];
%n=sqrt(u/a^3);
%qdot=qdot * ((n*a)/sqrt(1-e^2));
%R3w=[cos(w) sin(w) 0; -sin(w) cos(w) 0; 0 0 1];
%Rqx=R3w*R1*R3;
%Rxq=Rqx';

%XXX=Rxq*q;
%XXXdot=Rxq*qdot;
%XXX
%XXXdot
```