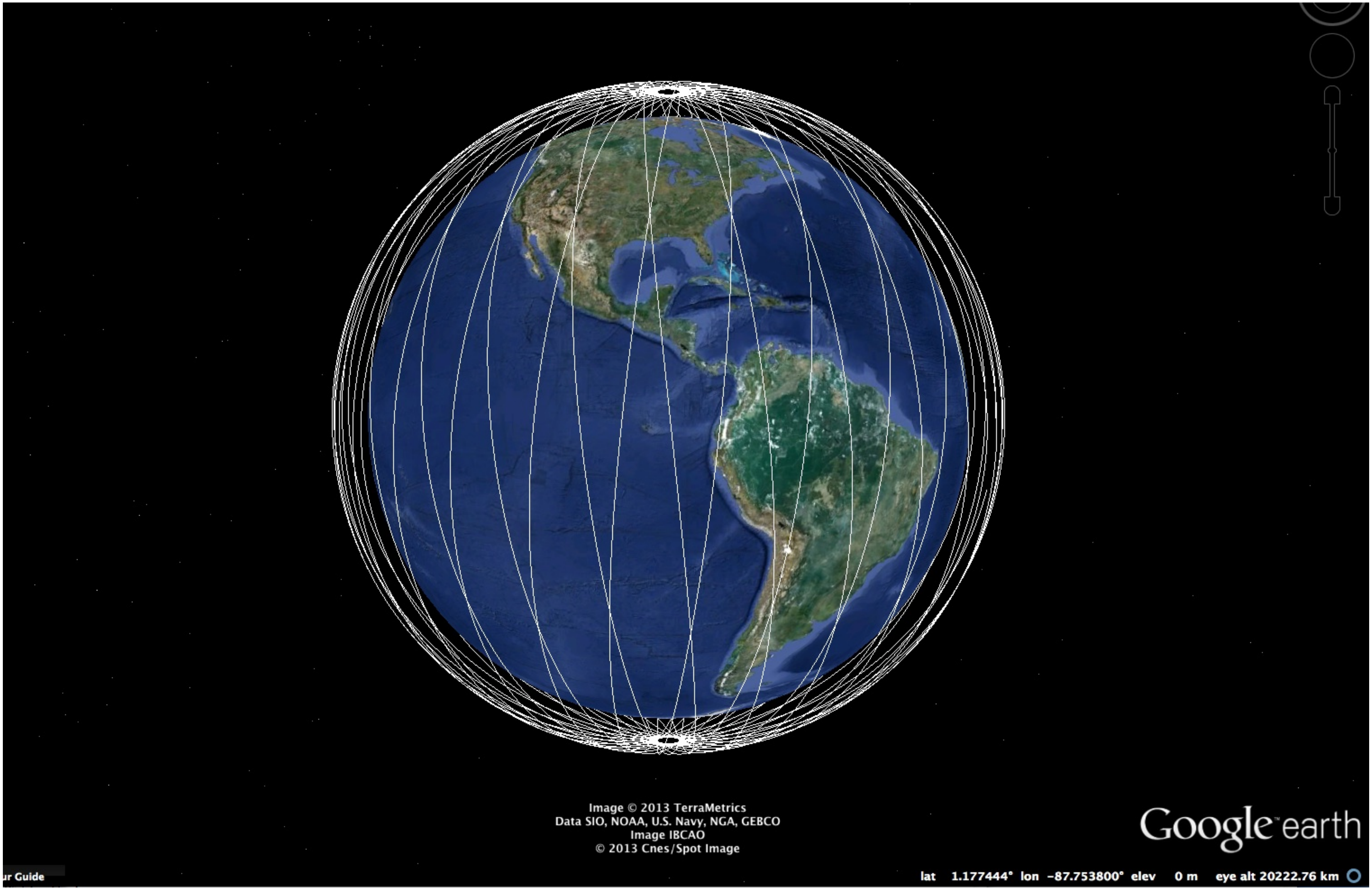
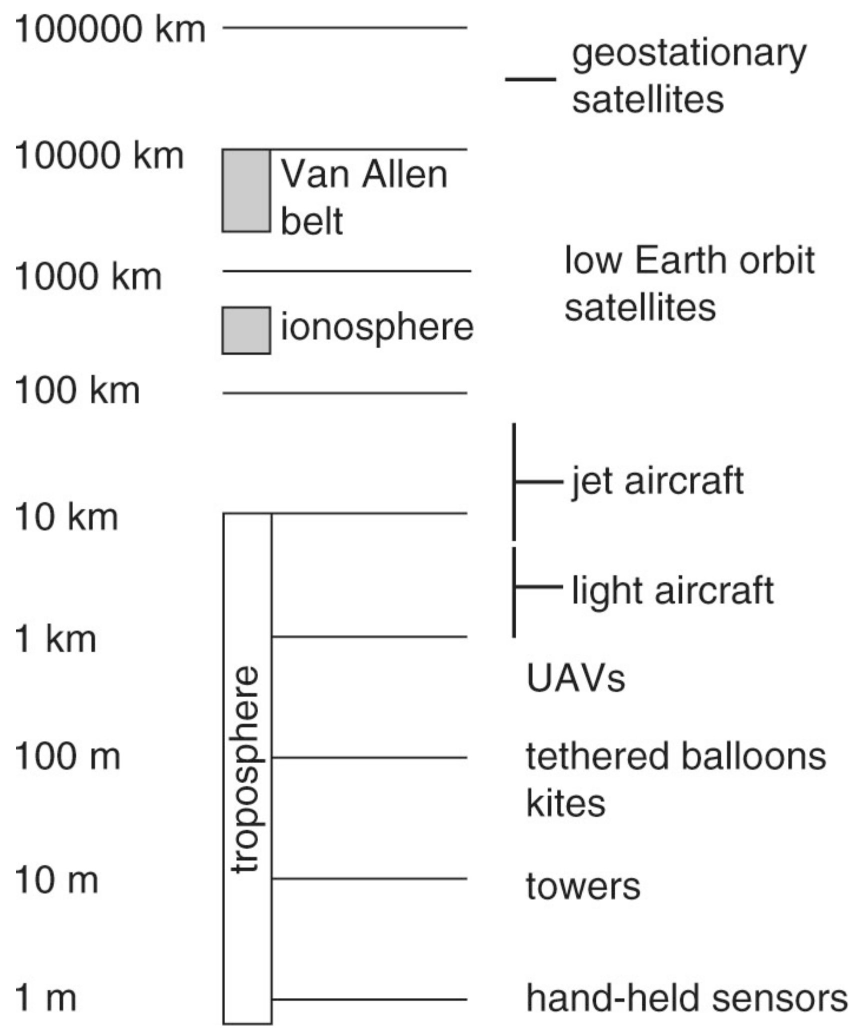
Platforms for Remote Sensing – Rees, Ch. 10

David Sandwell, April, 2014

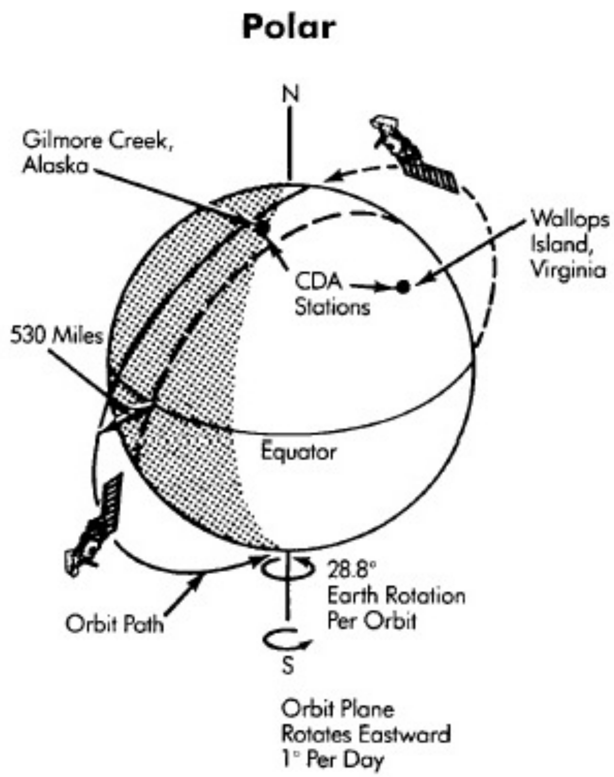


Altitudes of various remote sensing platforms



REES_Fig 10.01

low and high orbits



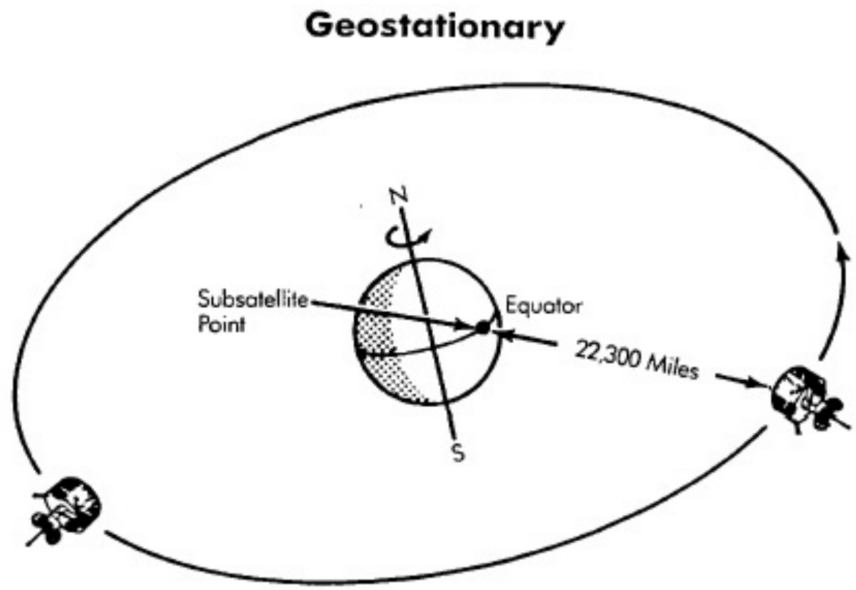
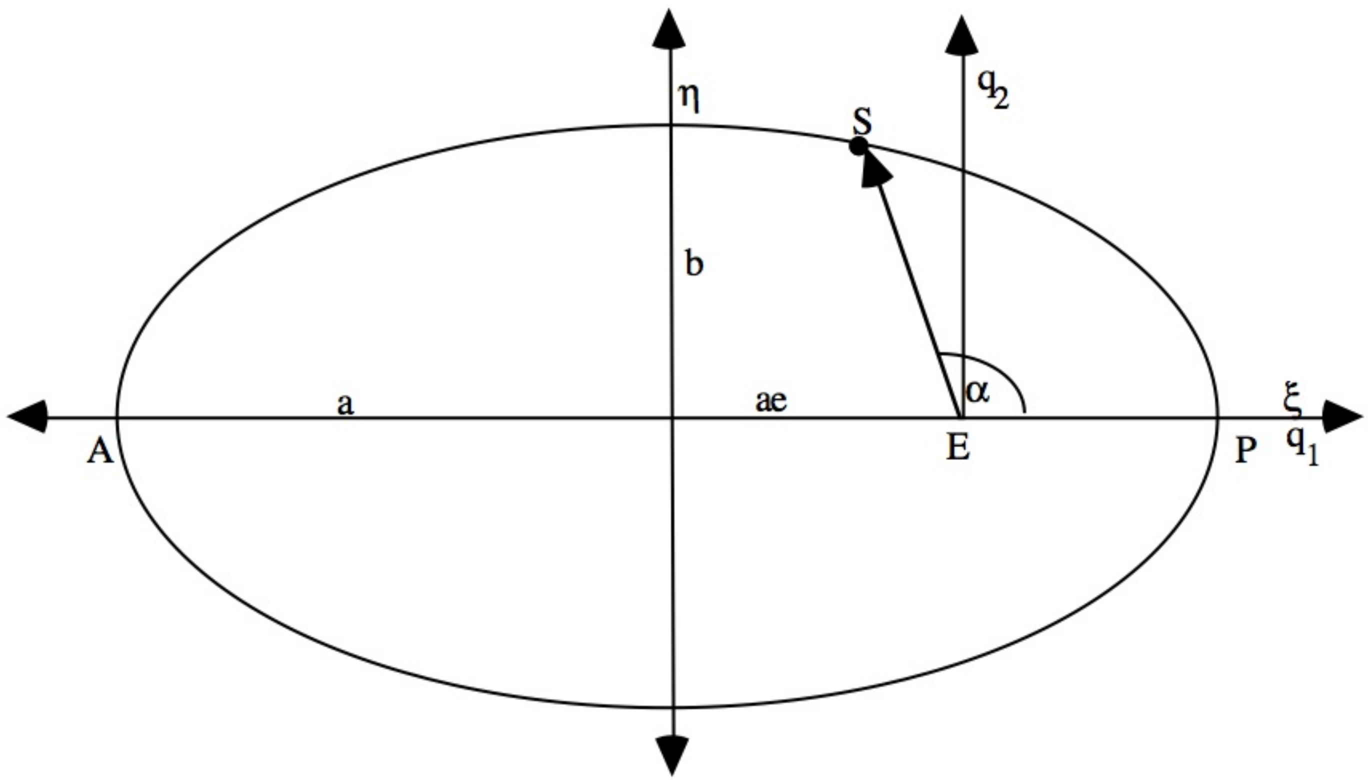


Figure 4. Opposite and above: polar and geostationary orbits for NOAA satellites. Note that the polar orbit rotates one degree per day; this is to make it synchronous with the sun. The geostationary satellite stays continuously above one spot on Earth.

NOTES ON BB

http://topex.ucsd.edu/rs/orbits.pdf

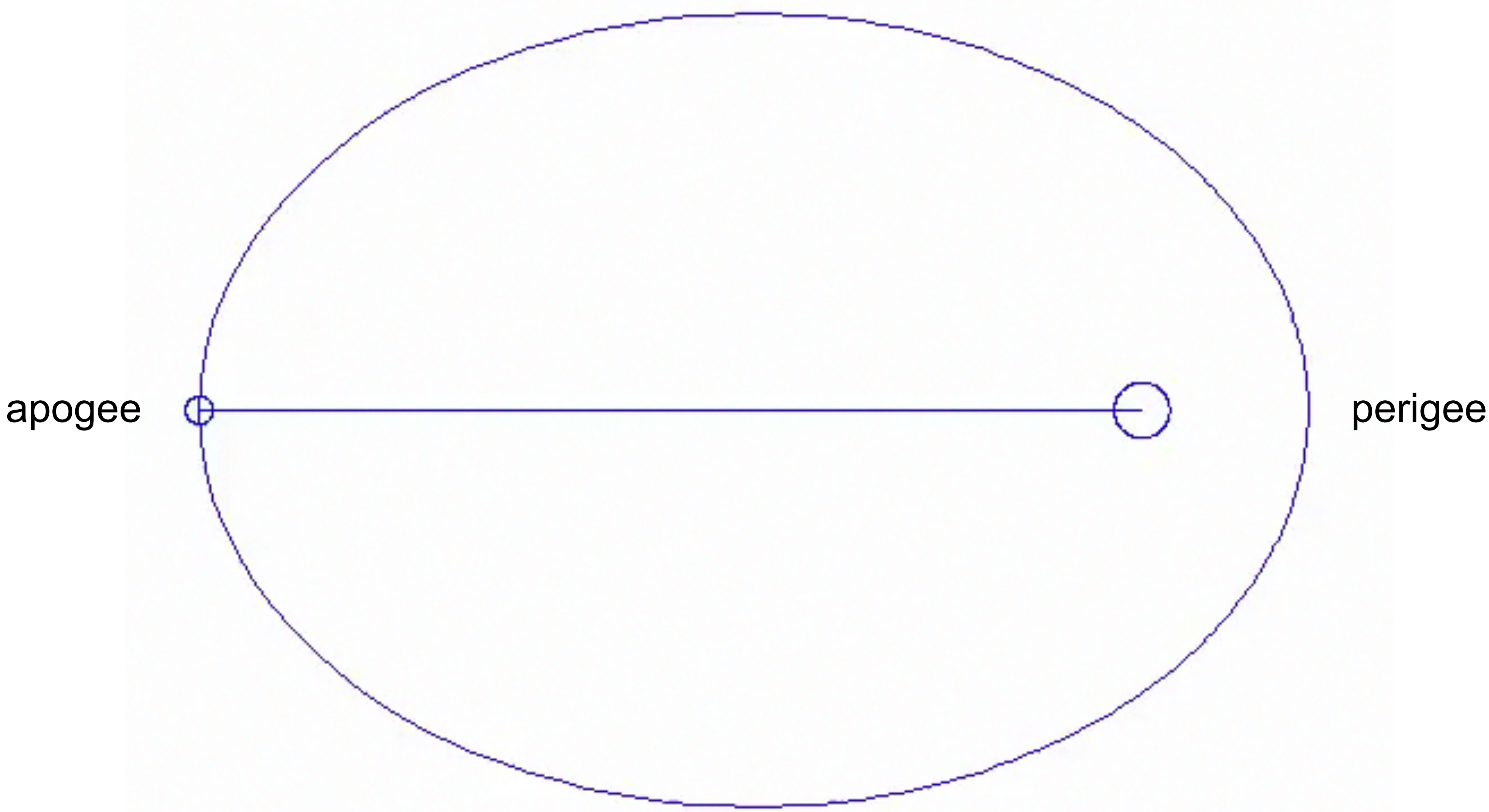
geometry of orbit plane



S	-	satellite
E	-	Earth
A	-	apogee
P	-	perigee
a	-	semimajor axis
b	-	semiminor axis
e	-	eccentricity

$$e^2 = \frac{a^2 - b^2}{a^2}$$

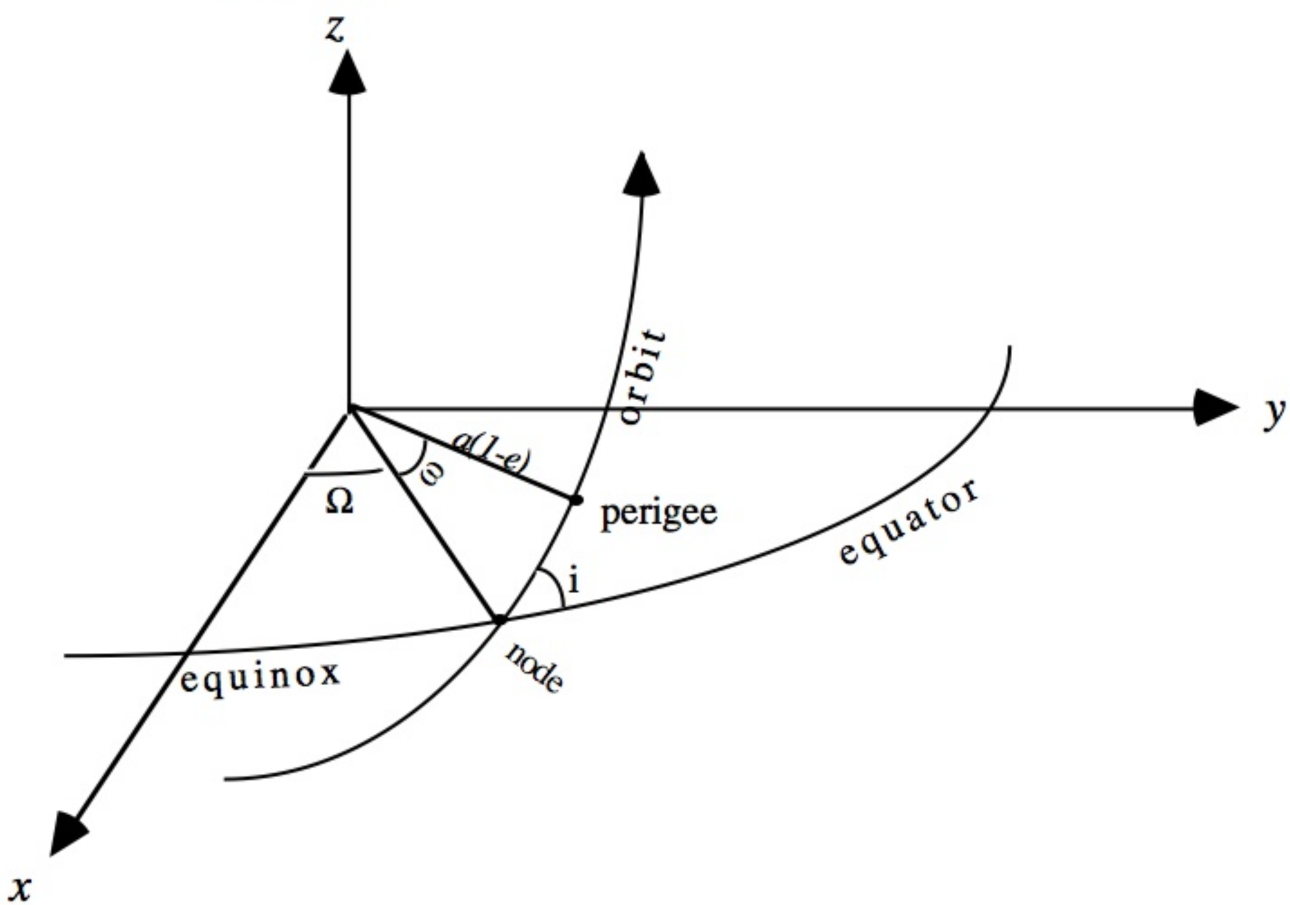
elliptical orbit motion



Orbital Geometry

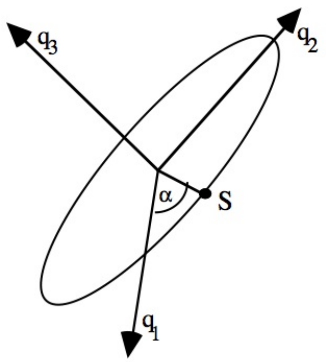
The ideal elliptical orbit is described by 6 Keplerian elements:

α	-	true anomaly (instaneous angle from satellite to perigee)
ω	-	argument of perigee
Ω	-	longitude of node
a	-	semimajor axis
e	-	eccentricity
i	-	inclination



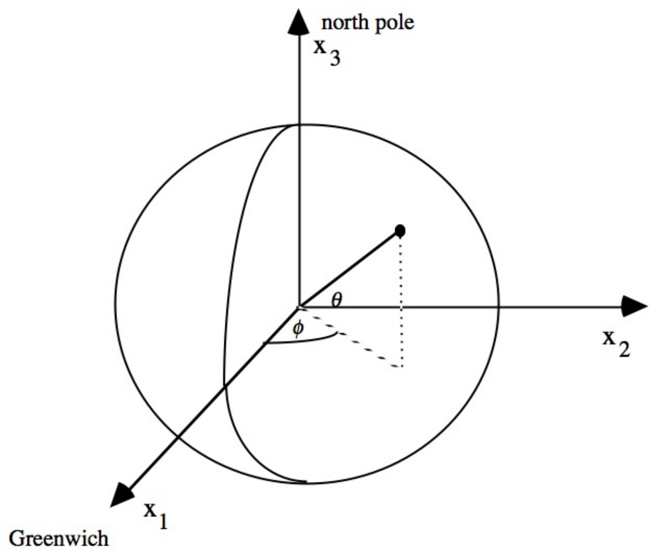
ground track of circular orbit

satellite frame



$$q_1 = \cos \omega_s t \; = \cos \alpha$$
$$q_2 = \sin \omega_s t \; = \sin \alpha$$
$$q_3 = 0$$

Earth-fixed frame



$$\theta(t) \;=\; \sin^{-1}\left[\sin \omega_s t \;\; \sin i\right] \qquad\qquad t(\theta) \;=\; \omega_s^{-1} \sin^{-1}\left[\frac{\sin \theta}{\sin i}\right]$$

$$\phi(t) \;=\; \tan^{-1}\left[\frac{-\sin \omega_e' t \;\; \cos \omega_s t \; + \cos \omega_e' t \;\; \sin \omega_s t \;\; \cos i}{\cos \omega_e' t \;\; \cos \omega_s t \; + \sin \omega_e' t \;\; \sin \omega_s t \;\; \cos i}\right] + \phi_o$$

$$x_1 = \cos \theta \cos \phi$$
$$x_2 = \cos \theta \sin \phi$$
$$x_3 = \sin \theta$$

Two rotations are needed to align the satellite frame **q** to the Earth-fixed frame **x**. First, the **q**-frame is rotated by an angle -*i* about the q_1 axis to account for the inclination of the orbit plane with

1) Zero Inclination Orbit $i = 0$ $\theta = 0$ and the orbit precession frequency is zero.

Using formulas for the sin and cosine of sums of angles (e.g., sin (a-b) = sin a cos b - cos a sin b) one can simplify the longitude function to the obvious result

$$\phi(t) = tan^{-1}\left[\frac{sin\,(\omega_s - \omega_e)t}{cos\,(\omega_s - \omega_e)t}\right] = (\omega_s - \omega_e)t$$

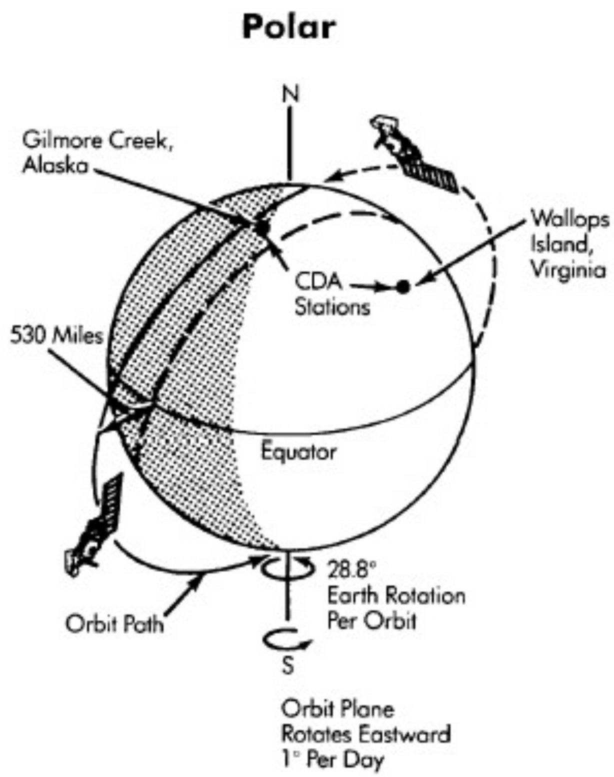
2) Geostationary Orbit $i = 0, e = 0, a_e = 6378135\text{m}$

Again the the satellite orbit frequency matches the Earth rotation rate $\omega_S = \omega_e$. From the equation above we have an expression for the orbit frequency for a flattened Earth.

$$\omega_s = \left(\frac{GM}{a^3}\right)^{1/2}\left[1 - \frac{9J_2}{2}\left(\frac{a_e}{a}\right)^2\right]^{-1} \qquad \omega_e = 2\pi/86146$$

The radius of the requires orbit is $a = 42170$ km or about 6.6 times the Earth radius. Usually this type of orbit is used for communications or for monitoring the weather patterns from a global

low and high orbits



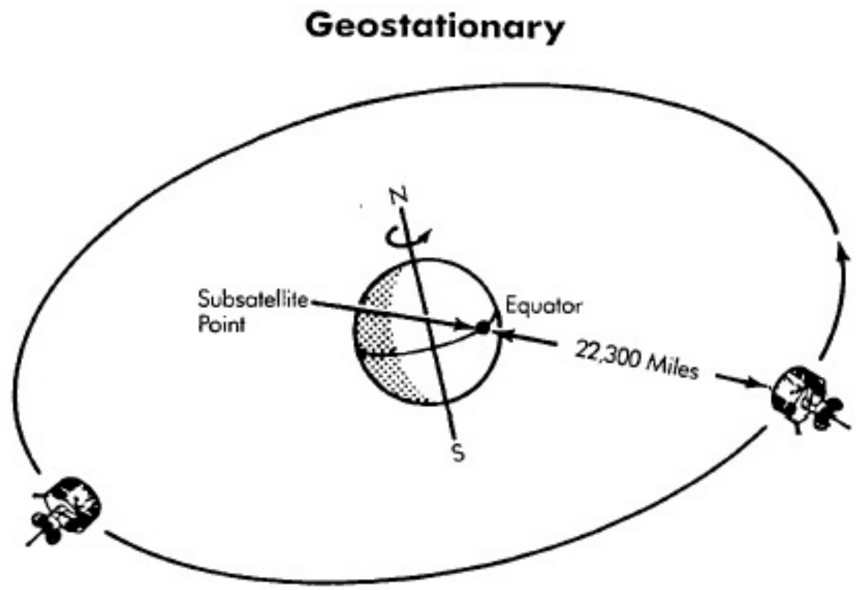
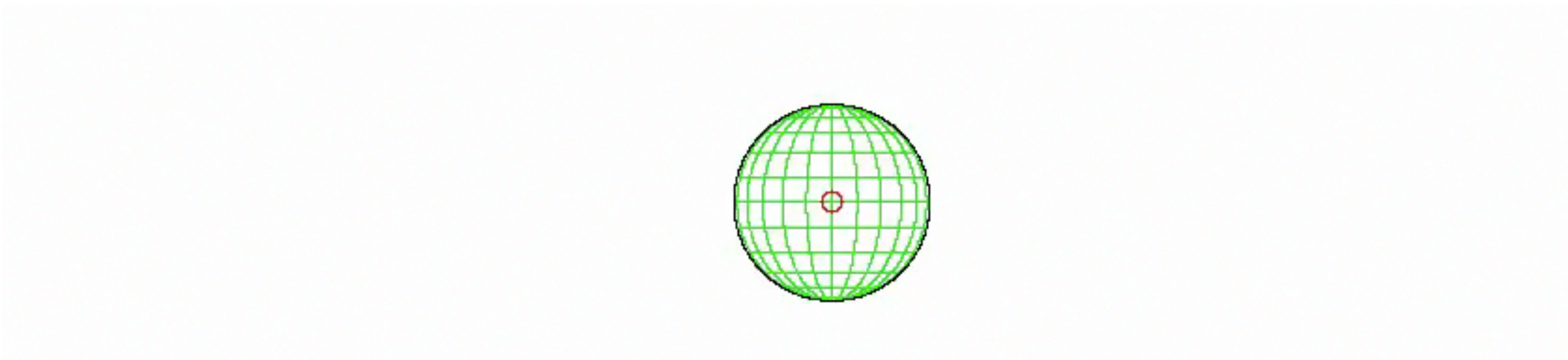


Figure 4. Opposite and above: polar and geostationary orbits for NOAA satellites. Note that the polar orbit rotates one degree per day; this is to make it synchronous with the sun. The geostationary satellite stays continuously above one spot on Earth.

geostationary orbit

orbit period = 1 day



3) Geosynchronous Orbit $i \neq 0, e = 0, \omega_S = \omega_e$

With a non-zero inclination, this orbit can cover higher latitudes but is spends only 1/2 of its time in the correct hemisphere. Inserting these parameters into the above ground track equations and neglecting the precession frequency of the orbit plane one obtains

$$\theta(t) = \sin^{-1}\left[\sin \omega_e t \ \sin i\right]$$

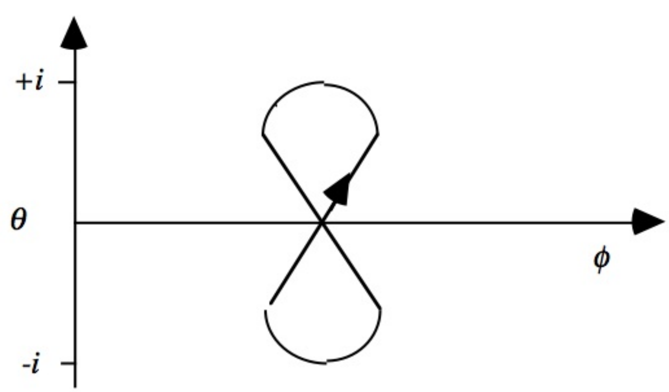
$$\phi(t) = \tan^{-1}\left[\frac{\cos \omega_e t \ \sin \omega_e t \ (\cos i - 1)}{\cos^2 \omega_e t \ + \cos i \ \sin^2 \omega_e t}\right]$$

Now consider the case of small inclination so cos $i \cong 1 - i^2/2$. The denominator is about 1 and the numerator can be simplified so the approximate results for longitude and latitude versus time are

$$\phi(t) = \tan^{-1}\left[\frac{i^2}{2}\frac{1}{2} \sin 2\omega_e t\right] \cong \frac{i^2}{4} \sin 2\omega_e t$$

$$\theta(t) = \sin^{-1}\left[\sin \omega_e t \ \sin i\right] \cong i \sin \omega_e t$$

The latitude varies as a sine wave with a frequency of ω_e while the longitude varies like a sine wave with a frequency of $2\omega_e$. At $t = 0$ both the latitude and longitude are zero. The ground track of the orbit follows a figure 8.

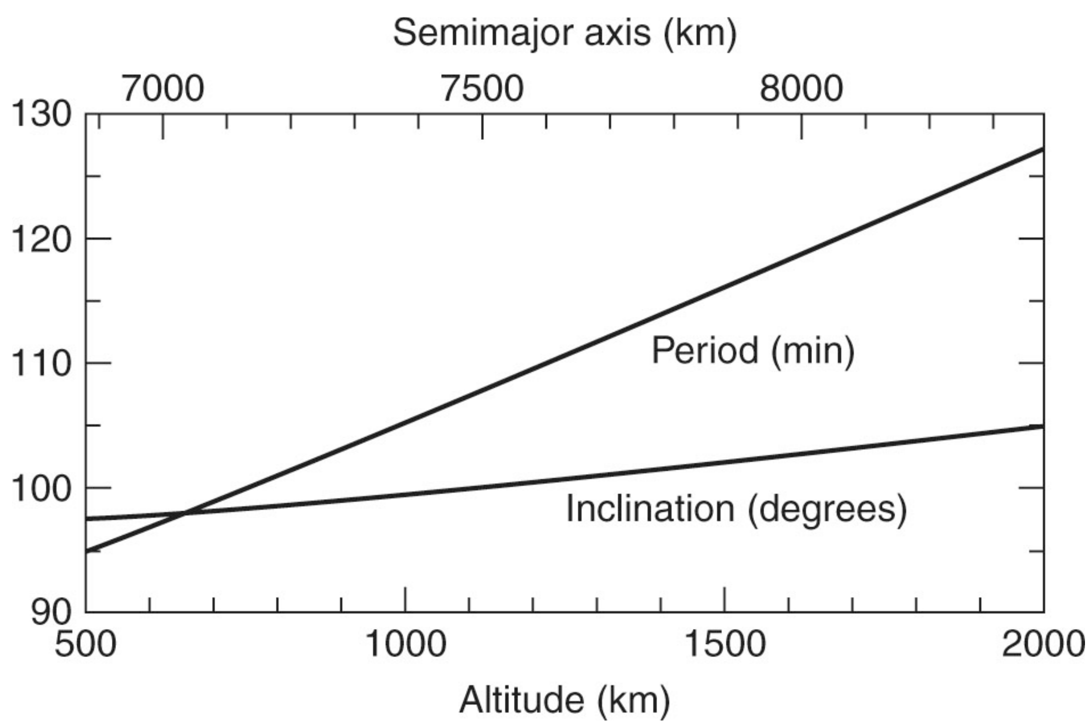


4) Sun-synchronous Orbit

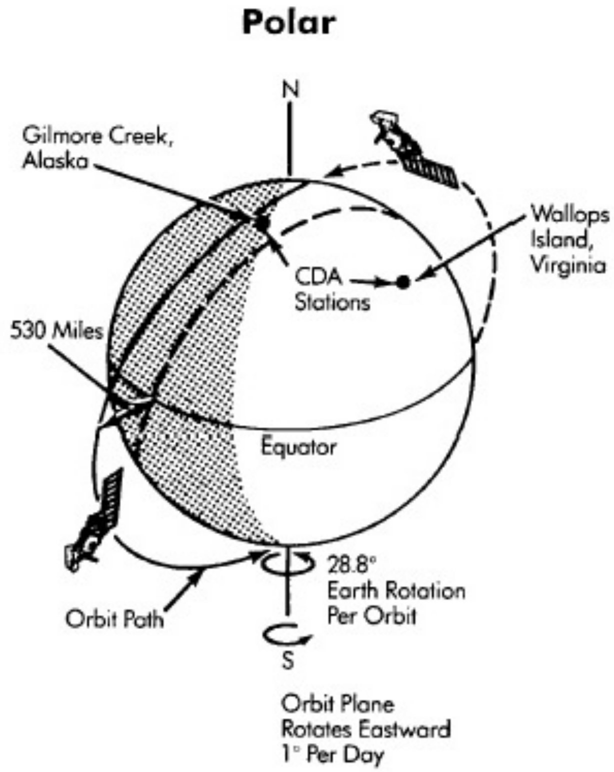
For many remote sensing applications it is important to have the ascending node pass over the equator at the same local time. To create a sun-synchronous orbit, the plane of the orbit must precess in a prograde direction with a period of exactly 1 year.

$$\omega_n = 2\pi/(365.25 \times 86400) = 1.991 \times 10^{-7} \text{ s}^{-1}$$

Remember $\frac{\omega_n}{\omega_s} = -\frac{3J_2}{2}\left(\frac{a_e}{a}\right)^2 \frac{\cos i}{(1-e^2)^2}$ so prograde precession requires $i > 90°$. Thus the orbital inclination is dictated by the orbital altitude. For example, a sun-synchronous orbit with an orbital radius of $a = 7878$ km (altitude 1500 km) must have an inclination of 102°.



low and high orbits



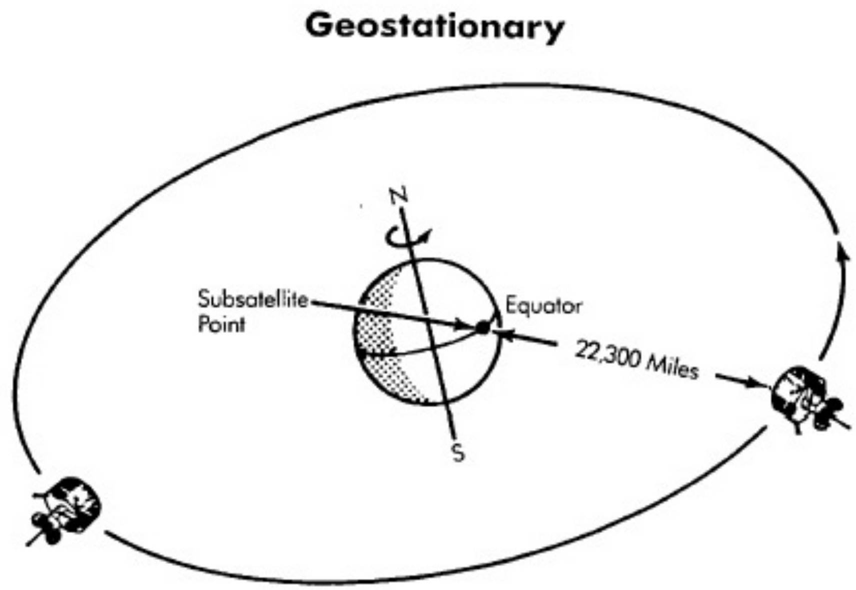
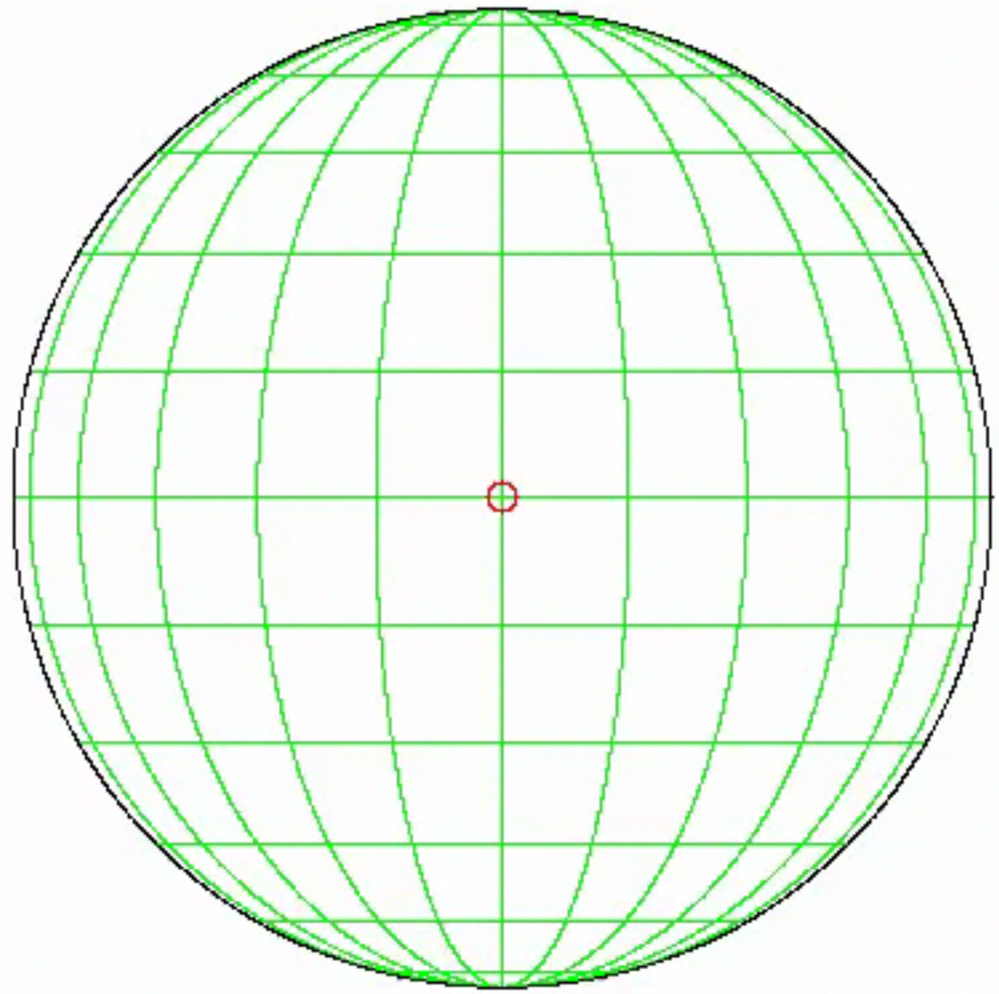


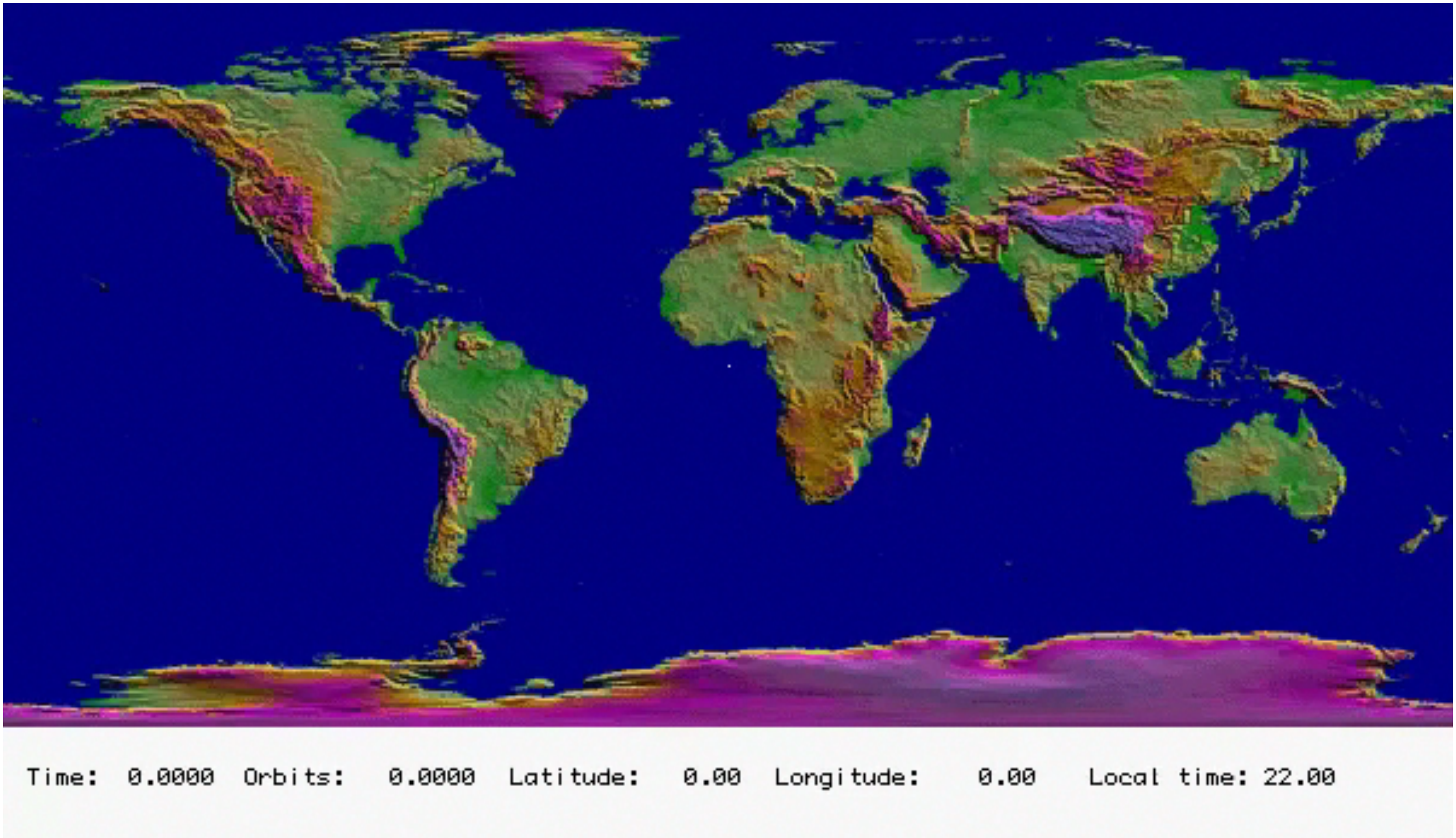
Figure 4. Opposite and above: polar and geostationary orbits for NOAA satellites. Note that the polar orbit rotates one degree per day; this is to make it synchronous with the sun. The geostationary satellite stays continuously above one spot on Earth.

sun synchronous orbit



precession of orbit plane

Landsat ground tracks



6) Exact Repeat Orbits

For oceanographic applications, where one wants to observe changes in ocean topography, the orbit must retrace its ground track so the large permanent signals associated with the Earth's gravity field can be subtracted out. To accomplish this there must be an integer relationship between the orbit frequency and the rotation rate of the Earth relative to the precessing orbit plane.

$2\pi/(\omega_e - \omega_n)\, n_1$ = time for n_1 rotations of the Earth relative to the orbit plane

$2\pi/\omega_e\ n_2$ = time it takes the satellite to complete n_2 orbits

$$\frac{2\pi}{\omega_e - \omega_n} n_1 = \frac{2\pi}{\omega_s} n_2 \Rightarrow \frac{\omega_e - \omega_n}{\omega_s} = \frac{n_1}{n_2}$$

Geosat Example,

$\omega_e = 7.292\text{x}10^{-5}\ \text{s}^{-1}$

$\omega_n = 4.144\text{x}10^{-7}\ \text{s}^{-1}$

$\omega_s = 1.041\text{x}10^{-3}\ \text{s}^{-1}$

$$\frac{\omega_e - \omega_n}{\omega_s} = \frac{17}{244}$$

5) Orbits Tuned for Ocean Altimetry

Criteria:

a) Ascending and descending tracks should cross at a high angle to resolve both components of sea surface slope.

b) One should be aware of the aliasing of Lunar and Solar tides into the altimeter profile.

c) One should choose a repeat cycle that will optimize spatial and temporal coverage.

d) High latitude coverage may be desired for ice altimetry.

tide		period - hours	period - days	phase shift after 17.05 days
M2	principal lunar	12.421	.5175	32.95 = 20°
K^1	lini-solar	23.934	.997	34°
S2	principal solar	12.00	.500	36°
O1	diurnal lunar	25.819	1.076	54°

How to compute an orbital track - 1

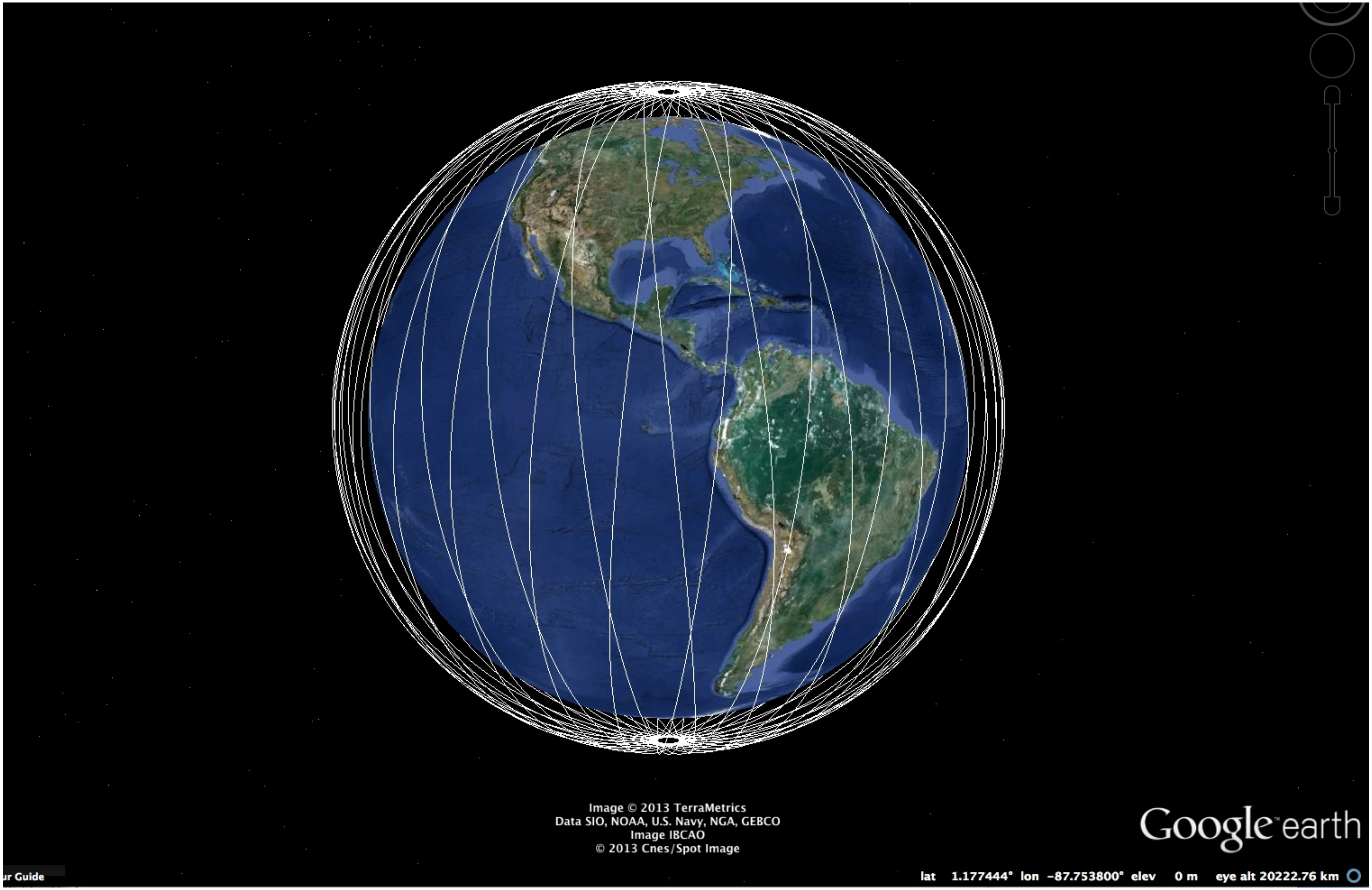
MAIN PROGRAM

```
%
%  SIO135/236 Demo of Ground Track for Circular Orbit
%
%  This code simulates a ground track for a satellite in a circular orbit about
%  an elliptical earth.  The formulas are derived in the class notes.
%
      clear;
      clf;
      flat = 1/298.257223560;
      tmax=86400*2.;
      t=0:10:tmax;
      we=2.*pi/86164.1;
%
%   cryosat
%
      [we,wn,ws,rinc]=setcnst(8);
      [rlon,rlatg] = trajec(t,we,wn,ws,rinc,flat);
      figure(5)
      plot(rlon,rlatg,'.');title('cryosat ');axis([-180,180,-90,90])
      pause
%
%  save a track for Google Earth
%
out=[rlon',rlatg'];
save -ascii cryosat.xy out

%
```

SET THE CONSTANTS

```
function[we,wn,ws,rinc]=setcnst(isat)
%
%  set constants and orbit parameters
%
%     input:
%     isat    satellite
%       1     geos-3
%       2     seasat
%       3     geosat
%       4     topex
%       5     ers-1
%       6     radarsat
%       7     JASON/GM
%       8     cryosat
%
%    output:
%      we      - earth rotation frequency 2*pi/86164
%      wn      - satellite precession frequency
%      ws      - satellite orbit frequency
%      rinc    - orbital inclination
%
we=2.*pi/86164.1;
rad=pi/180.;
%
% set the orbital period and inclination of each satellite
%
Tss =
[6029.928,6037.377,6037.55,6745.72,6035.928,6045.420,6745.72,5965.86];
rincs =
[114.98,108.0584,108.0584,66.01,98.5557,98.581755,66.01,92.00067];
wss = 2.*pi./Tss;
%
% set the precession frequency for each satellite
%
wns=zeros(8);
wns(1) = we/176.;
wns(2) = we/176.;
wns(3) = we/175.968;
wns(4) = we-wss(4)*10./127.;
wns(5) = we-wss(5)*35./501.;
wns(6) = we-wss(6)*24./243.;
wns(7) = we-wss(7)*419./5321.;
wns(8) = we-wss(8)*369./5344.;
%
%   load the output values
%
ws = wss(isat);
wn = wns(isat);
rinc = rincs(isat)*rad;
return
```

How to compute an orbital track



How to compute precise orbits from elements

PRECISE STATE VECTORS FOR ENVISAT SPACED AT 60 SECOND INTERVALS

TIME	X	Y	Z	DX/DT	DY/DT	DZ/DT
21:55:28.00	+4925775.296	-0672489.270	+5149599.921	-5468.045305	-1230.624128	+5056.89226
21:56:28.00	+4588023.337	-0743509.493	+5442738.786	-5786.560443	-1135.486760	+4711.22472
21:57:28.00	+4231855.868	-0808605.919	+5714579.288	-6081.687559	-1033.262456	+4347.17957
21:58:28.00	+3858712.626	-0867371.390	+5964061.354	-6352.228261	-0924.559337	+3966.18750
21:59:28.00	+3470102.184	-0919436.302	+6190212.706	-6597.087114	-0810.020339	+3569.74320
22:00:28.00	+3067595.664	-0964470.578	+6392152.503	-6815.275024	-0690.319610	+3159.39884
22:01:28.00	+2652820.268	-1002185.442	+6569094.621	-7005.912367	-0566.159017	+2736.75872
22:02:28.00	+2227452.596	-1032334.971	+6720350.620	-7168.232800	-0438.264632	+2303.47344
22:03:28.00	+1793211.746	-1054717.441	+6845332.311	-7301.586442	-0307.382928	+1861.23307
22:04:28.00	+1351852.262	-1069176.424	+6943553.921	-7405.442085	-0174.276681	+1411.76019
22:05:28.00	+0905156.961	-1075601.647	+7014633.841	-7479.388480	-0039.720920	+0956.80327

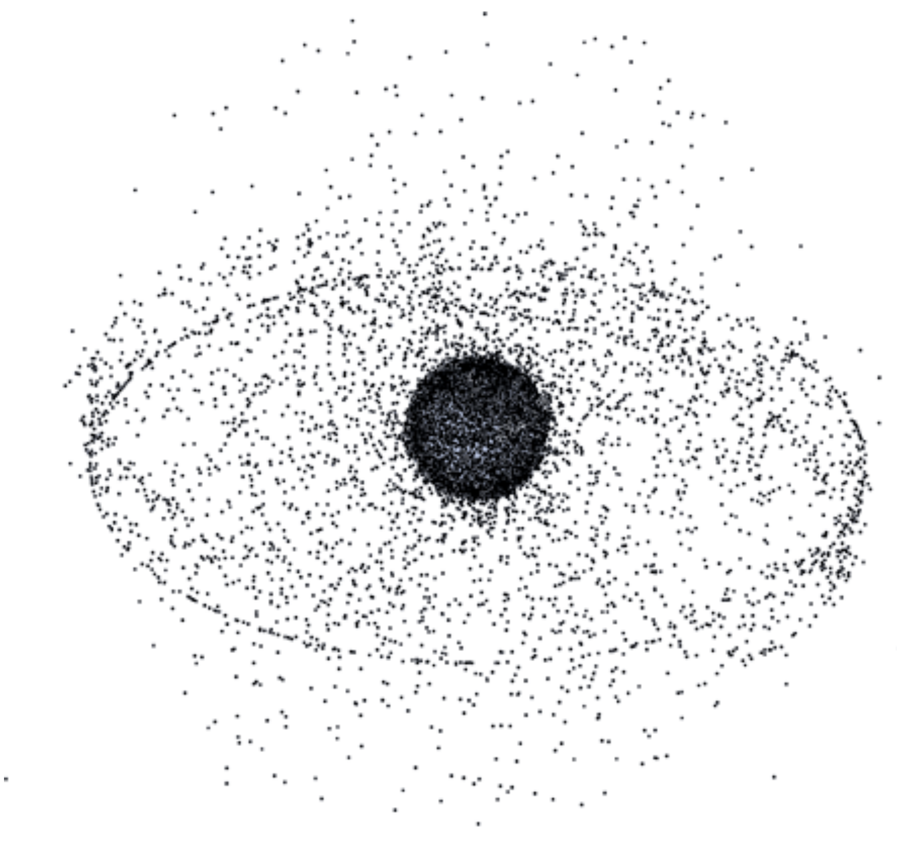
Orbital elements are based on precise GPS tracking and orbit dynamic modeling.

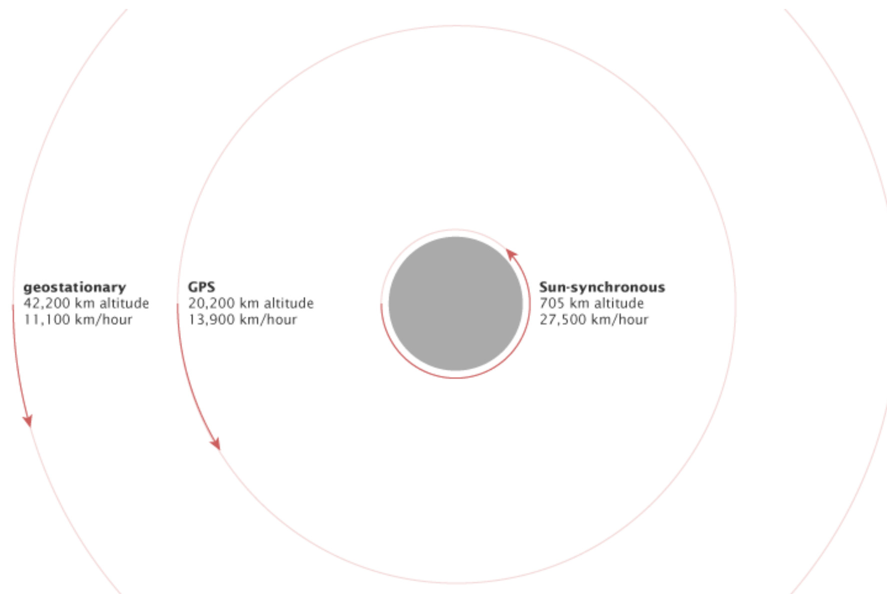
C-PROGRAM TO INTERPOALTE STATE VECTORS

```
/*---------------------------------------------------------------*/
void interpolate_ALOS_orbit(struct ALOS_ORB *orb, double *pt, double *p,
                double *pv, double time, double *x, double *y, double *z, int *ir)
{
/* ir;                  return code            */
/* time;               seconds since Jan 1     */
/* x, y, z;            position               */
int     k, nval, nd;

        nval = 6; /* number of points to use in interpolation */
        nd = orb->nd;

        if (debug) fprintf(stderr," time %lf nd %d\n",time,nd);

        /* interpolate for each coordinate direction    */

        /* hermite_c c version                         */
        for (k=0; k<nd; k++) {
                p[k] = orb->points[k].px;
                pv[k] = orb->points[k].vx;
                }

        hermite_c(pt, p, pv, nd, nval, time, x, ir);

        for (k=0; k<nd; k++) {
                p[k] = orb->points[k].py;
                pv[k] = orb->points[k].vy;
                }
        hermite_c(pt, p, pv, nd, nval, time, y, ir);

        for (k=0; k<nd; k++) {
                p[k] = orb->points[k].pz;
                pv[k] = orb->points[k].vz;
                }
        hermite_c(pt, p, pv, nd, nval, time, z, ir);

}
```

Platforms for Remote Sensing – Rees, Ch. 1

~18,000 man-made objects orbiting the Earth





http://earthobservatory.nasa.gov/Features/OrbitsCatalog/page3.php