Samara State Aerospace University

Manual for First Semester Course Project

Samara 2015

Table 1: Keplerian orbit elements: Satellite position

a	semi major axis	size and shape of orbit
e	eccentricity	
ω	argument of perigee	the orbital plane in the apparent system
Ω	right ascension of ascending node	
i	inclination	
μ	mean anomaly	position in the plane

1. Project Contents

The present course project combines useful and basic insight to GNSS orbits with the necessary MATLAB coding. Emphasis is on visualization of GNSS orbits and making the student familiar with orbit descriptions as given by almanacs and ephemerides. *Any observer needs to know the time windows in which the satellites are visible at a given location.*

Much of the theoretical background needed you may find in Chapter 9 of Borre & Strang (2012). However, this manual details central issues which may be difficult to find in a textbook and offers some help for how to code. In Table 1 we collect the six Keplerian elements which uniquely describe a course satellite orbit.

GNSS orbital information is provided in slightly different forms. In this project we restrict ourselves to details of GPS.

2. The GPS Orbital Information

The orbital information comes in two versions, as almanacs and as ephemerides. The former contains the Keplerian elements only, while the latter one is augmented with additional parameters which allow for more precise computation of satellite positions. An ephemeris takes 21 parameters.

They are listed below in the sequence in which they are used in all my software developed since 1997, especially in the Easy-suite. An alternative sequence is used in the Receiver INdependent EXchange (RINEX) format, see Gurtner & Estey (2007). However, we maintain our own defined sequence which originally was introduced in our and other non-commercial softwares:

```
GM = 3.986005e14; % earth's universal gravitational
% parameter m^3/s^2
Omegae_dot = 7.2921151467e-5; % earth rotation rate, rad/s

%  Units are either seconds, meters, or radians
%  Assigning the local variables to eph
svprn   =  eph(1);
af2     =  eph(2);
M0      =  eph(3);
roota   =  eph(4);
deltan  =  eph(5);
ecc     =  eph(6);
omega   =  eph(7);
cuc     =  eph(8);
cus     =  eph(9);
crc     =  eph(10);
crs     =  eph(11);
i0      =  eph(12);
idot    =  eph(13);
cic     =  eph(14);
cis     =  eph(15);
Omega0  =  eph(16);
Omegadot=  eph(17);
toe     =  eph(18);
af0     =  eph(19);
af1     =  eph(20);
toc     =  eph(21);
```

Table 2: Students and Ephemerides/Almanacs

Name	Almanac	φ [°]	λ [°]	h [m]	Cut-off [°]
Sergio	*.15n	40	0	0	10
Nelson	*.15n	50	10	0	10
Jaime	*.15n	50	20	0	15
Gabriel	*.15n	20	0	0	10
Jessica	*.15n	45	70	0	10
Darwin	*.15n	55	80	0	15
Luis	*.15n	60	10	0	5
Shervin	*.15n	40	30	0	10
Sergey	*.15n	30	30	0	20
Ëkaterina	*.15n	20	30	0	10
Vladimir	*.15n	0	40	0	5

The matrix eph has 21 rows keeping the Kepler elements and other information. Any almanac file may be run through rinexe where the matrix eph is created with 21 rows and columns for all satellites (PRN); in the ephemeris version they may appear repeatedly in several columns while for an almanac each satellite only appears once. So you may formally create an ephemeris from an almanac by setting all parameters not defined in the almanac to zero in the ephemeris.

We emphasize that this course project only outputs graphical results, and no numerical results.

3. How to Get Started

In your moodle you will find one ephemerides file (zimm0750.15n). This ephemerides file is created at Zimmerwald, Germany. You may download your own ephemerides file from the internet by googling brdc0750.15n. Of course, the position for your course project as indicated in Table 2 must match the location where the ephemerides file was created. The

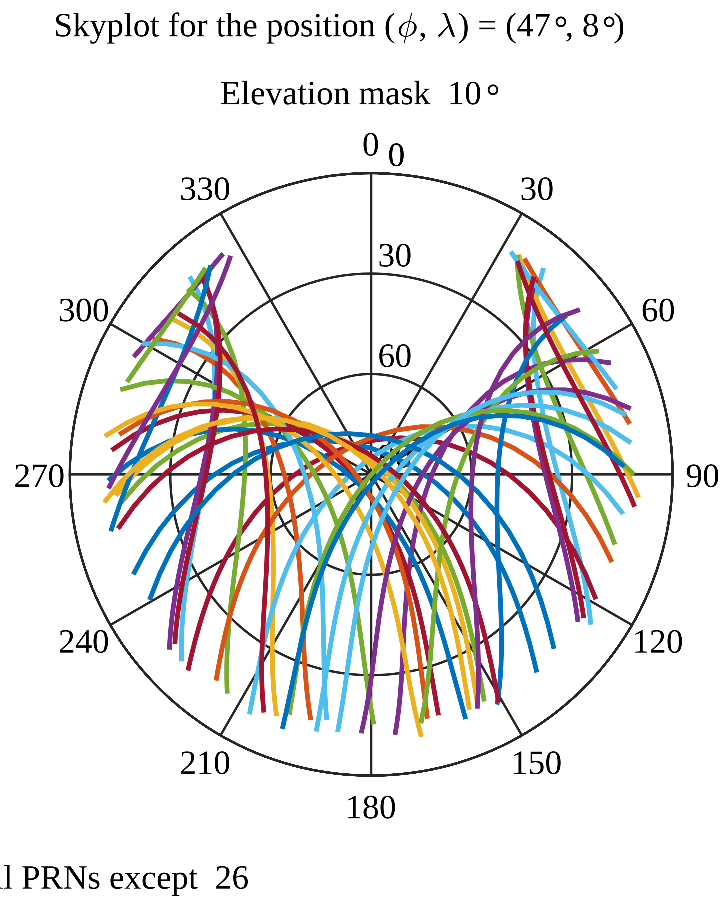

Figure 1: Sky plot including all GPS satellites for a period of 24 hours at a given position. The elevation mask is given in Table 2

ephemerides file only contains satellites visible at that location. If you want to get free of this restriction you must switch to use almanacs. Remember almanac files are universal.

This course project is closely connected to the easy11. So you may start from the easy11 version you find in your moodle and modify this file. Already now I can tell, that this code needs to be modified to work. The idea is that you *learn* from this project!

Your almanac file may be used as input to the rinexe-file to create the matrix eph. This matrix has 21 rows and an unknown number of columns.

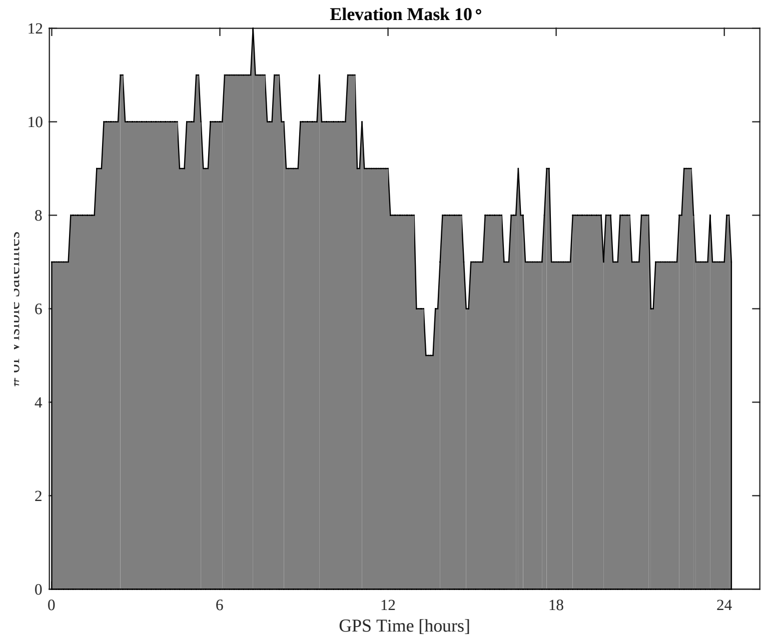

Figure 2: Number of visible satellites

The matrix can be stored in MATLAB's internal binary format in the file mynav.nav, say.

In addition we need to fix the value of the *elevation mask* or the *cut-off angle*. It is the angle between the horizon and the elevation angle from which we start tracking satellites. Observations closer to the horizon are useless, as they are disturbed by too much tropospheric delay. Your specific value for the elevation angle is also indicated in Table 2.

4. Coordinate Transformation

In general we use a Cartesian 3-D coordinate system for GNSS computations. So we need to transform the given coordinates (φ, λ, h) into (X, Y, Z). This algorithm is only described to you in the next semester, but can be found as equation (3.3) in Borre & Strang (2012). The M-file frgeod performs this computation. An explanation of the code is given in Section 3.1.3 in the mentioned textbook.

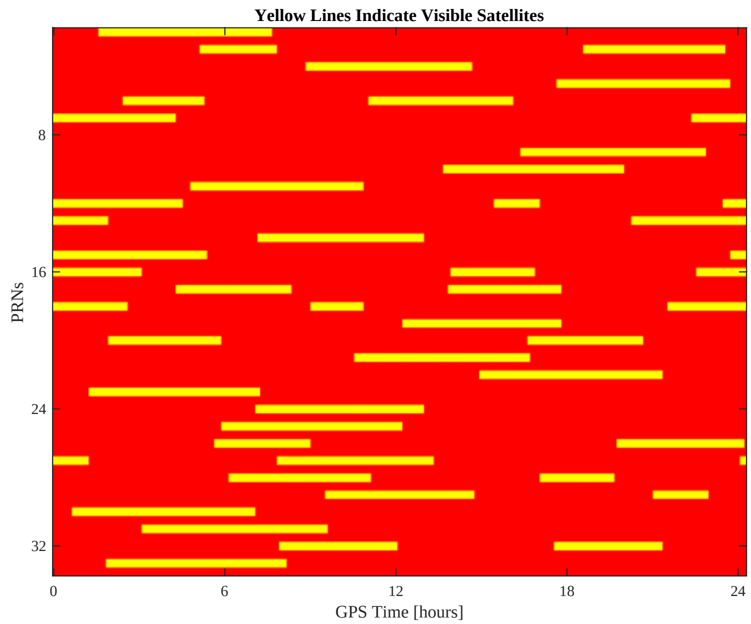

Figure 3: Time period when the visible satellites are 10° or higher above the horizon

For a and finv you must use the values for the semi-major axis a and the inverse flatting $1/f$ as given in the Interface Control Documents for GPS.

Now we may compute the coordinates of satellite sat for a given time of week time and with given eph. The statement S = satpos(time,eph(:,sat)); will do the computation. It is a complicated code which we discussed in detail in the present course.

5. Computation of Azimuth and Elevation

Again we are in a situation where you must rely on simpel transformation formulas which only will be described to you in the next semester. They can be found on page 275 in Borre & Strang (2012).

Compute (az, el) for each satellite for every 10 or 15 minutes over a period of 24 hours.

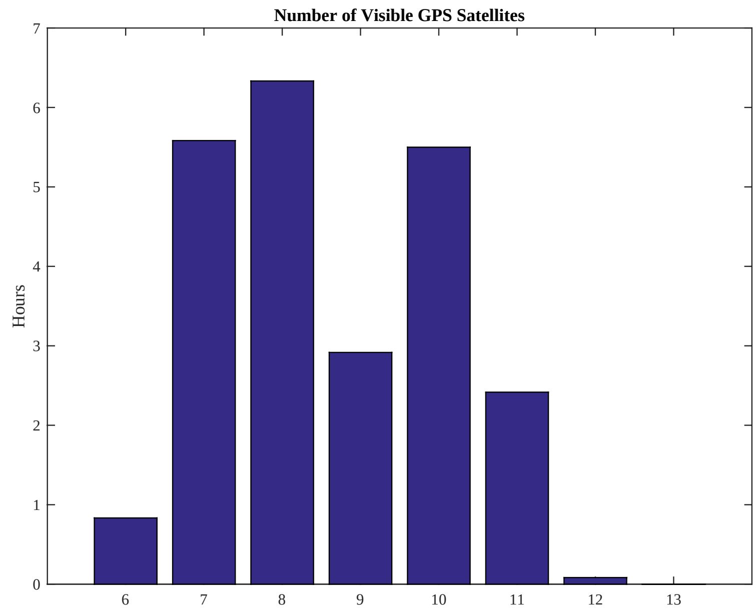

Figure 4: Number of satellites visible in number of hours

6. Specifications of the Plots

Now you can call satpos and compute the (X, Y, Z) position of any satellite at any time. in which the ephemeris is valid.

You make a skyplot for GPS satellites valid for the values given in Table 2, see Figure ??. The time interval between positions can be 10 or 15 minutes in an interval of 24 hours.

The skyplot can be a little tricky to create. The polar coordinates are $(az, 90° - el)$. So the annotation in the plot is reverse from zenith (the center) to the horizon (outermost circle). Ask me for help if necessary.

The next plot shall tell a user how many satellites he can find above his horizon and delimited by the elevation mask. You must count the number of satellites above the elevation mask for every 10 minutes, say. Time is along the horizontal axis and the number of visible satellites along the vertical axis. The result looks like Figure 2. You may experiment with various values of time intervals. Do time resolutions of 20 or 30 minutes lead to too crude intervals?

A different plot showing visibility is given as Figure 3. Finally the number of visible satellites are shown in Figure 4.

You are most welcome to make graphical experiments with different line widths, colors, etc.

You also can investigate the influence of making plots for different sites like Samara, your own home, or $(\varphi, \lambda) = (80^\circ, 0^\circ)$. Finally the elevation mask can be selected as 0°, 5°, 10°, or 15°.

7. Deadline is December 20, 2015

The course-project report must be sent as a pdf file by e-mail to borre@gps.aau.dk not later than December 20, 2015.

References

Borre, Kai & Strang, Gilbert (2012). *Algorithms for Global Positioning.* Wellesley-Cambridge Press, Wellesley MA.

Gurtner, Werner & Estey, Lou (2007). *RINEX: The Receiver Independent Exchange Format.* `ftp://igscb.jpl.nasa.gov/igscb/data/format/rinex300.txt`, version 3.00 edition.