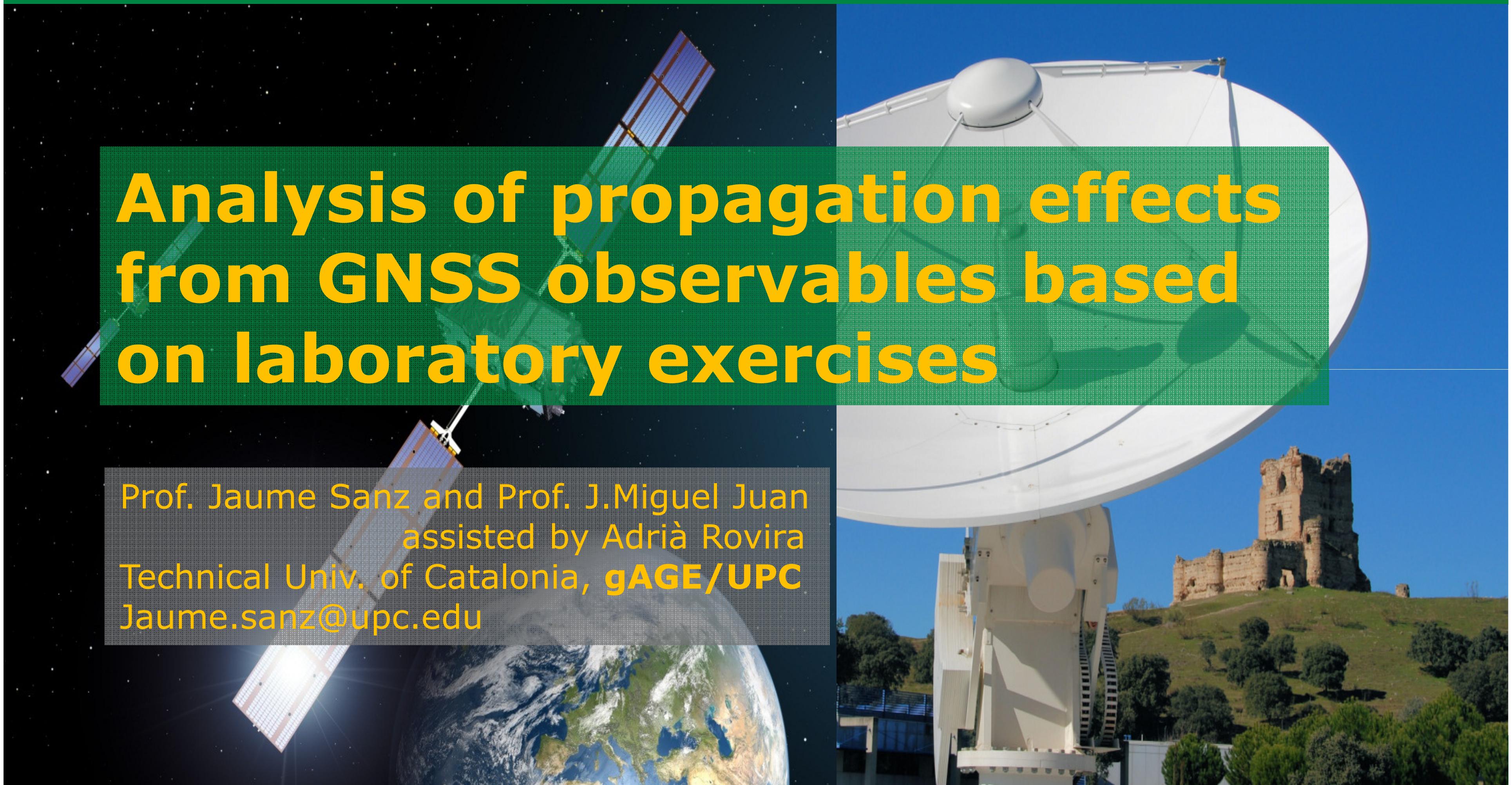
Analysis of propagation effects from GNSS observables based on laboratory exercises

Prof. Jaume Sanz and Prof. J.Miguel Juan
assisted by Adrià Rovira
Technical Univ. of Catalonia, **gAGE/UPC**
Jaume.sanz@upc.edu



Course on "Propagation effects, channel models and related error sources on GNSS"
ESAC, Madrid, Spain – 15-17 October 2012

Authorship statement

Overview

- **Introduction**

1. The gLAB tool suite
2. Examples of GNSS Data Processing using gLAB
3. Laboratory session organization

LABORATORY Session

- Starting up your laptop
- Basic: Introductory lab exercises: Iono & Posit, SF, storm,TIDs
- Medium: Laboratory Work Projects: LWP1, LWP2
- Advanced: Homework

Introduction

- **This practical lecture** is devoted to analysing and assessing different issues associated with GNSS signal propagation effects in the atmosphere.
- **The laboratory exercises** will be developed with actual GPS measurements, and processed with the ESA/UPC GNSS-Lab Tool suite (gLAB), which is an interactive software package for GNSS data processing and analysis.
- **All software tools** (including gLAB) and associated files for the laboratory session are included in the USB stick delivered to those who attend the lecture.
- **The laboratory session** will consist of a set of exercises organized in three different levels of difficulty (Basic, Medium and Advanced). A set of introductory examples range from a first glance assessment of the ionosphere effects on single frequency positioning, and Zenith Tropospheric Delays estimate to showing different perturbation effects in the ionosphere (Solar Flair, Halloween storm, TIDs). Electron density profiles (Ne) retrieval, bending effects analysis (phase excess rate depicture) are analysed in detail in two Laboratory Work Projects. Finally the code-carrier ionosphere divergence on single-frequency smoothed codes is proposed as homework.
- **The target** is to provide the participants with a wide range of selected exercises to choose from, according their interests and their level of knowledge of these topics.

Overview

1. Introduction
- **The gLAB tool suite**
2. Examples of GNSS Data Processing using gLAB
3. Laboratory session organization

LABORATORY Session

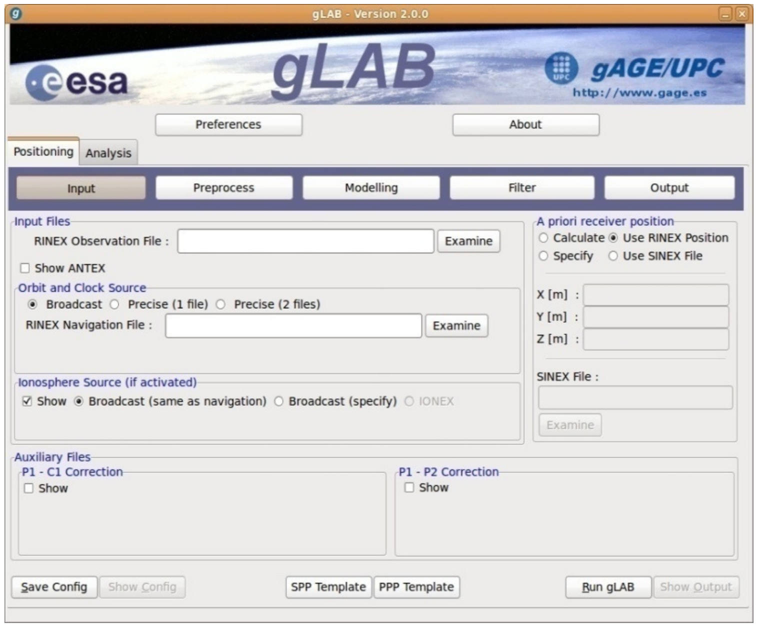
- Starting up your laptop
- Basic: Introductory lab exercises: Iono & Posit, SF, storm,TIDs
- Medium: Laboratory Work Projects: LWP1, LWP2
- Advanced: Homework

The gLAB Tool suite

- The GNSS-Lab Tool suite (gLAB) is an interactive multipurpose educational and professional package for GNSS Data Processing and Analysis.

- gLAB has been developed under the ESA Education Office contract N. P1081434.

- Main features:
  - High Accuracy Positioning capability.
  - Fully configurable.
  - Easy to use.
  - Access to internal computations.



The gLAB Tool suite

1. Students/Newcomers:

a. Easy to use: Intuitive GUI.

b. Explanations: Tooltips over the different options.

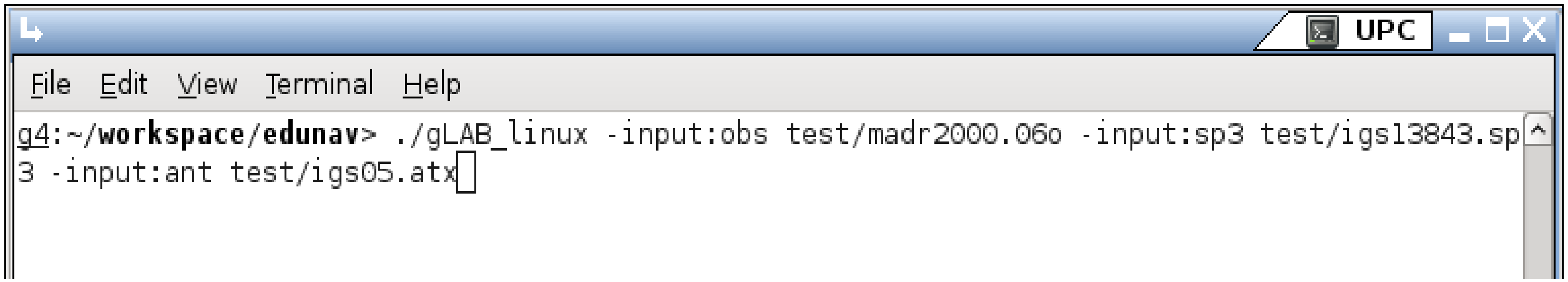
c. Guidelines: Several error and warning messages. Templates for pre-configured processing.

2. Professionals/Experts:

a. Powerful tool with High Accuracy Positioning capability.

b. Fast to configure and use: Templates and carefully chosen defaults.

c. Can be executed in command-line and **included in batch processing**.



The gLAB Tool suite

1. In order to widen the tool availability, gLAB Software has been designed to work in both Windows and Linux environments.

2. The package contains:
    - Windows binaries (with an installable file).
    - Linux .tgz file.
    - Source code (to compile it in both Linux and Windows OS) under an Apache 2.0 license.
    - Example data files.
    - Software User Manual.
    - HTML files describing the standard formats.

The gLAB Tool suite

- Read files capability:
  - RINEX observation v2.11 & v3.00
  - RINEX navigation message.
  - SP3 precise satellite clocks and orbits files
  - ANTEX Antenna information files.
  - Constellation status.
  - DCBs files.
  - GPS_Receiver_Type files.
  - SINEX position files.

- Pre-processing module:
  - Carrier-phase pre-alignment.
  - Carrier-phase / pseudo-range consistency check.
  - Cycle-slip detection (customizable parameters)
    - Melbourne-Wübbena.
    - Geometry-free CP combination.
    - L1-C1 difference (single frequency).
  - Pseudo-range smoothing.
  - Decimation capability.
  - On demand satellite enable/disable.
  - Elevation mask.
  - Frequency selection.
  - Discard eclipsed satellites.

- Modelling module:
  - Fully configurable model.
  - Satellite positions.
  - Satellite clock error correction.
  - Satellite movement during signal flight time.
  - Earth rotation during signal flight time.
  - Satellite phase center correction.
  - Receiver phase center correction. (frequency dependent).
  - Relativistic clock correction.
  - Relativistic path range correction.
  - Ionospheric correction (Klobuchar).
  - Tropospheric correction
    - Simple and Niell mappings.
    - Simple and UNB-3 nominals.
  - Differential Code Bias corrections.
  - Wind up correction.
  - Solid tides correction (up to 2nd degree).

The gLAB Tool suite

- Filtering module:
  - Able to chose different measurements to process (1 or more), with different weights. This design could be useful in future Galileo processing, where processing with different measurements may be desired.
  - Fixed or elevation-dependent weights per observation.
  - Troposphere estimation on/off.
  - Carrier-Phase or Pseudo-range positioning.
  - Static/Kinematic positioning (full Q/Phi/P0 customization).
  - Able to do a forward/backward processing.
  - Able to compute trajectories (no need for a priori position).

- Output module:
  - Cartesian / NEU coordinates.
  - Configurable message output.

- Other functionalities:
  - Computation of satellite coordinates and clocks from RINEX and SP3 files.
  - Satellite coordinates comparison mode. For instance RINEX navigation vs. SP3, or SP3 vs. SP3 (along-track, cross-track and radial orbit errors, clock errors, SISRE).
  - Show input mode. No processing, only parsing RINEX observation files.

- Current version allows full GPS data processing, and partial handling of Galileo and GLONASS data.
- Future updates may include full GNSS data processing.

GNSS learning material package

Includes three different parts, allowing participants to follow either a guided or a self-learning GNSS course:

1. GNSS Book: Complete book with theory, practical examples, and with a Laboratory course on GNSS Data Processing & Analysis [R-1].

2. gLAB tool suite: Source code and binary software files, plus configuration files, allowing processing GNSS data from standard formats. The options are fully configurable through a GUI.

3. gAGE-GLUE: Bootable USB stick with a full environment ready to use; based on LINUX (Ubuntu) OS.

Overview

1. Introduction
2. The gLAB tool suite
➢ **Examples of GNSS processing using gLAB**
4. Laboratory session organization

LABORATORY Session

- Starting up your laptop
- Basic: Introductory lab exercises: Iono & Posit, SF, storm,TIDs
- Medium: Laboratory Work Projects: LWP1, LWP2
- Advanced: Homework

Examples of GNSS Data Processing using gLAB

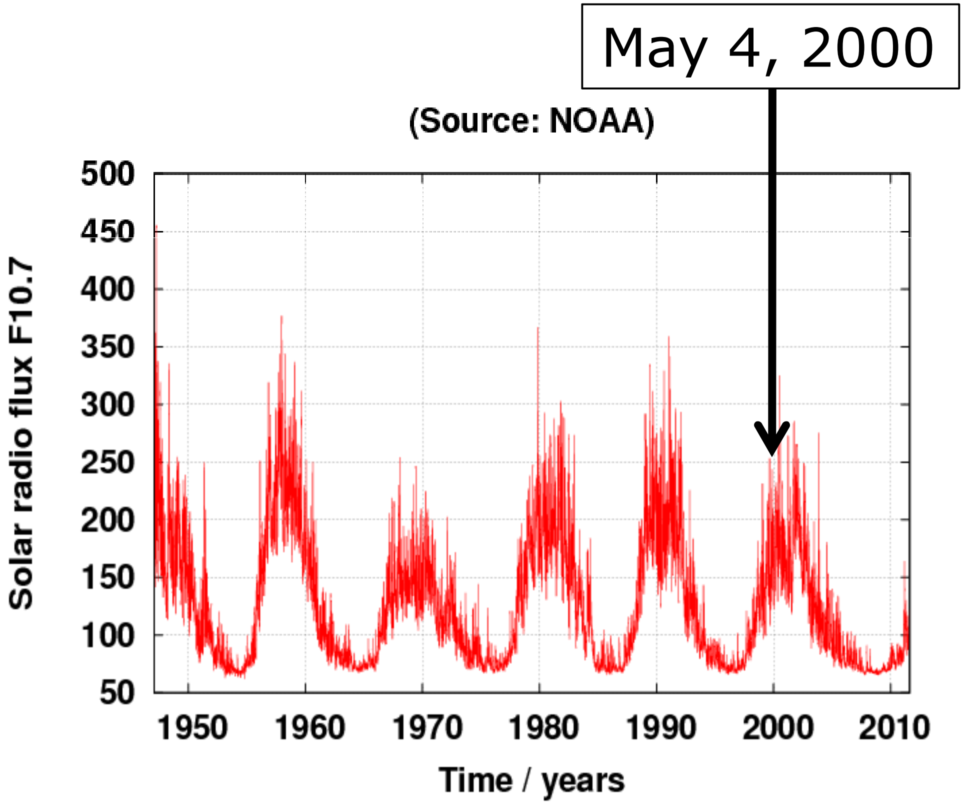
Example 1: Ionospheric effects on single frequency positioning.

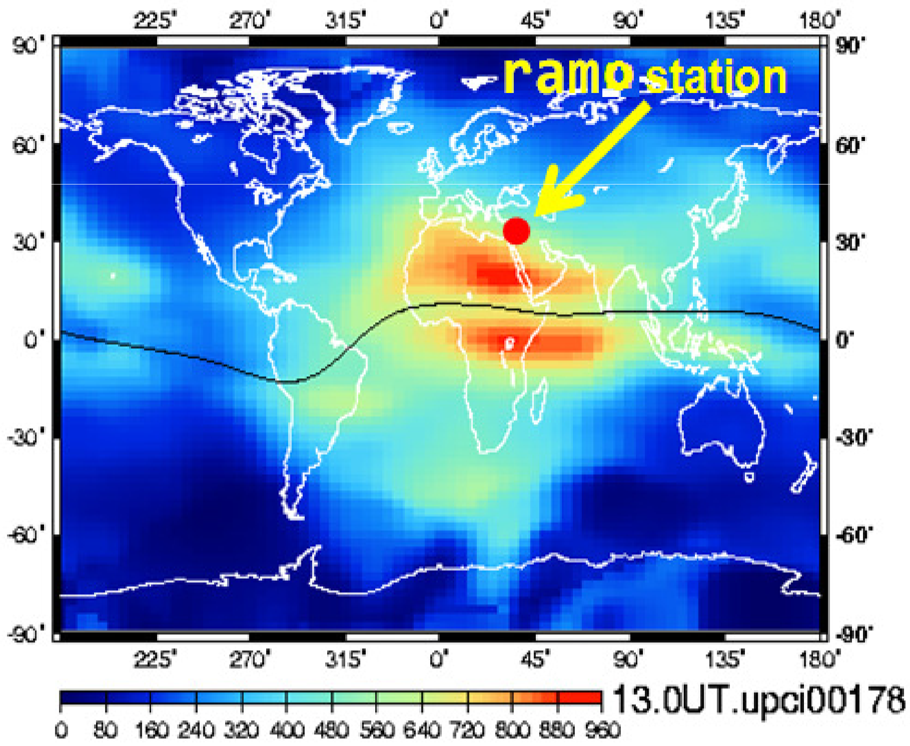
a. This exercise is devoted to analysing the effect of the ionospheric error in single frequency positioning. This is done both in the Signal-In-Space (SIS) and User Domains.

b. A receiver will be positioned in Standard Point Positioning (SPP) mode: a) with full modelling, b) neglecting the ionospheric correction.

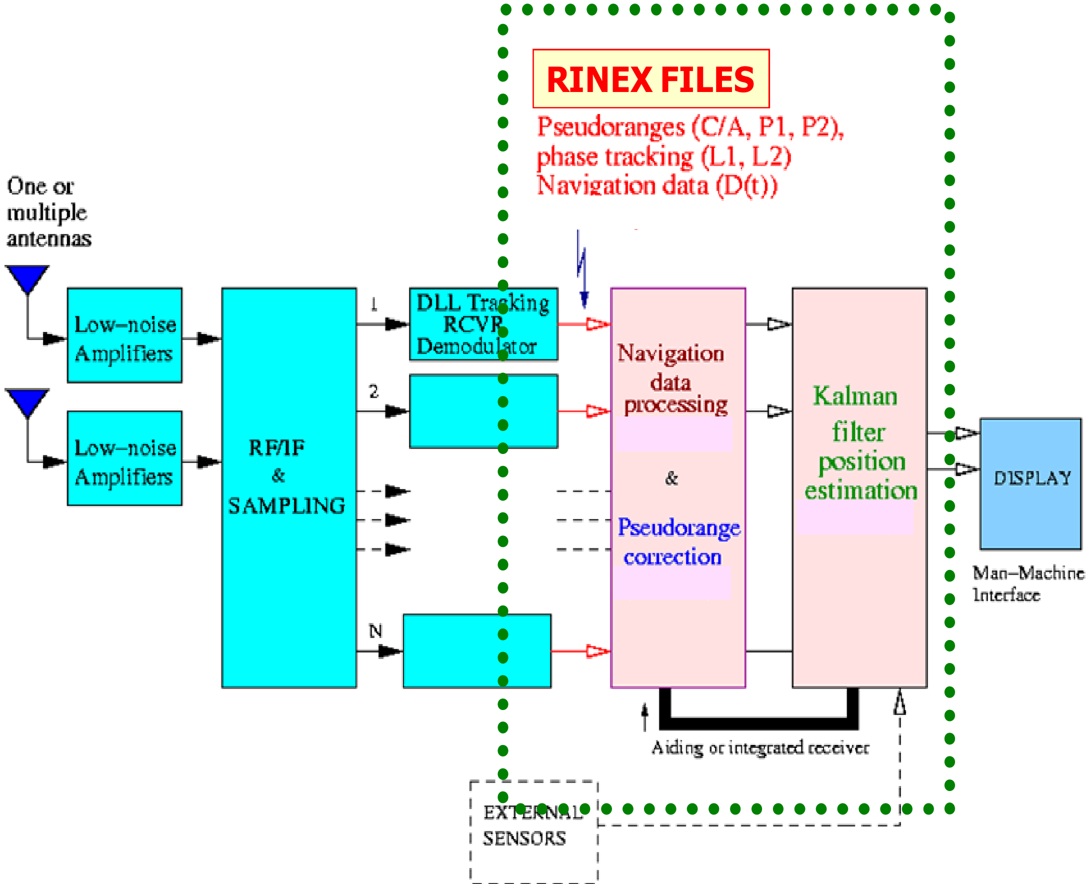
Example 1:
Iono effects on single freq. Positioning

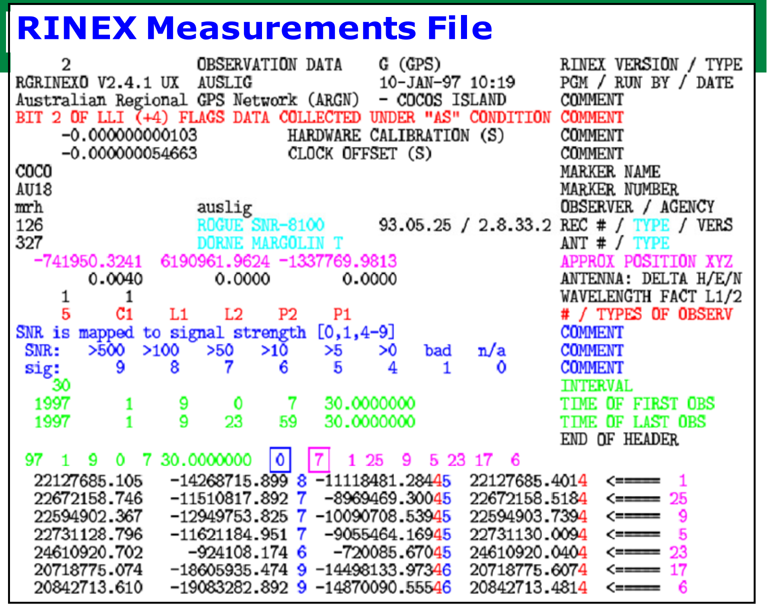
Data set: 24h data collected by the IGS permanent receiver "ramo" (Lon,Lat) on May 4th 2000.





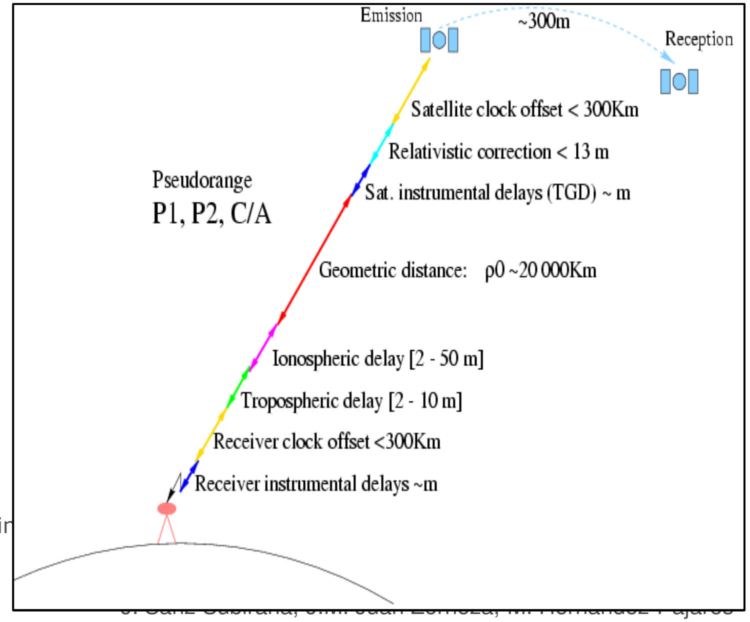
gLAB works after the correlator: The input data are code and carrier measurements and satellite orbits and clocks.





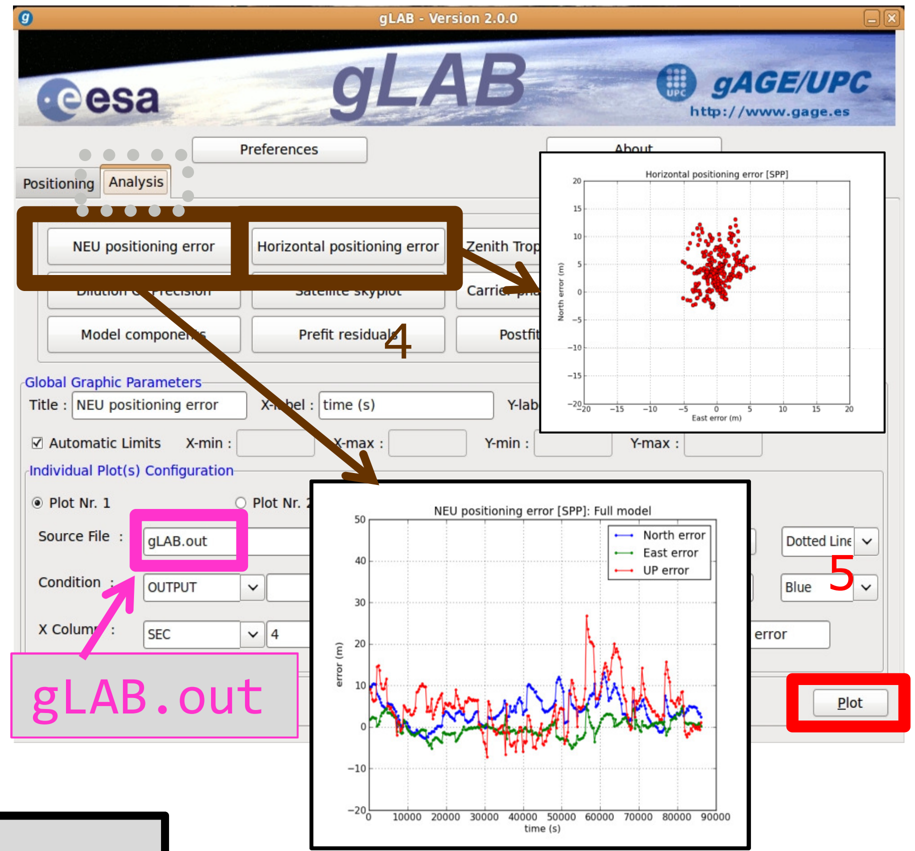
RINEX Measurements File

```
     2              OBSERVATION DATA    G (GPS)             RINEX VERSION / TYPE
RGRINEXO V2.4.1 UX  AUSLIG              10-JAN-97 10:19     PGM / RUN BY / DATE
Australian Regional GPS Network (ARGN) - COCOS ISLAND       COMMENT
BIT 2 OF LLI (+4) FLAGS DATA COLLECTED UNDER "AS" CONDITION COMMENT
     -0.000000000103          HARDWARE CALIBRATION (S)      COMMENT
     -0.000000054663          CLOCK OFFSET (S)              COMMENT
COCO                                                        MARKER NAME
AU18                                                        MARKER NUMBER
mrh                 auslig                                  OBSERVER / AGENCY
126                 ROGUE SNR-8100      93.05.25 / 2.8.33.2 REC # / TYPE / VERS
327                 DORNE MARGOLIN T                        ANT # / TYPE
  -741950.3241  6190961.9624 -1337769.9813                  APPROX POSITION XYZ
        0.0040        0.0000        0.0000                  ANTENNA: DELTA H/E/N
     1     1                                                WAVELENGTH FACT L1/2
     5    C1    L1    L2    P2    P1                        # / TYPES OF OBSERV
SNR is mapped to signal strength [0,1,4-9]                  COMMENT
 SNR:    >500  >100   >50   >10    >5    >0   bad   n/a     COMMENT
 sig:       9     8     7     6     5     4     1     0     COMMENT
    30                                                      INTERVAL
  1997     1     9     0     7   30.0000000                 TIME OF FIRST OBS
  1997     1     9    23    59   30.0000000                 TIME OF LAST OBS
                                                            END OF HEADER
 97  1  9  0  7 30.0000000  0  7  1 25  9  5 23 17  6
  22127685.105   -14268715.899 8 -11118481.28445  22127685.4014  <====  1
  22672158.746   -11510817.892 7  -8969469.30045  22672158.5184  <==== 25
  22594902.367   -12949753.825 7 -10090708.53945  22594903.7394  <====  9
  22731128.796   -11621184.951 7  -9055464.16945  22731130.0094  <====  5
  24610920.702     -924108.174 6   -720085.67045  24610920.0404  <==== 23
  20718775.074   -18605935.474 9 -14498133.97346  20718775.6074  <==== 17
  20842713.610   -19083282.892 9 -14870090.55546  20842713.4814  <====  6
```



Example 1:
Iono effects on single freq. Positioning

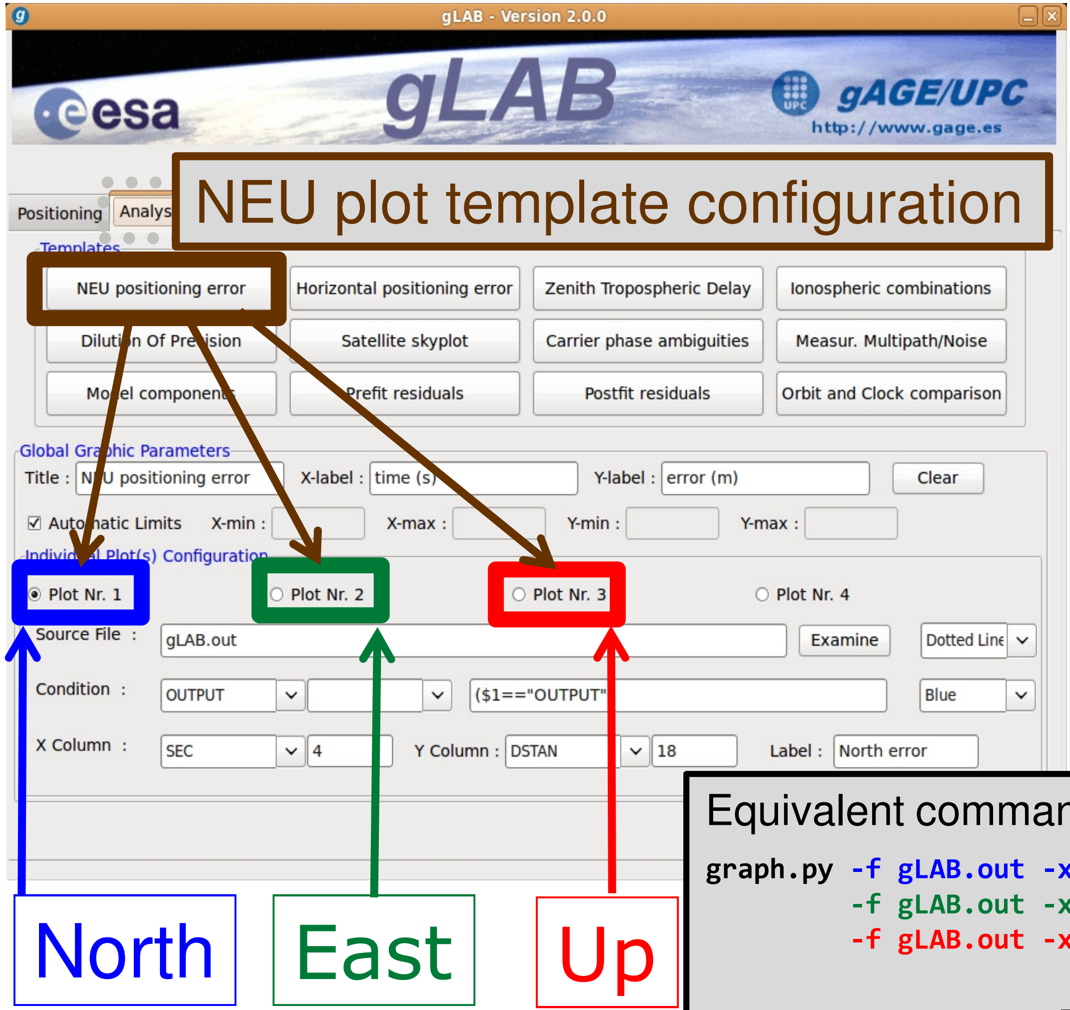
1. Compute SPP using files: ramo1250.00o,brdc1250.00n



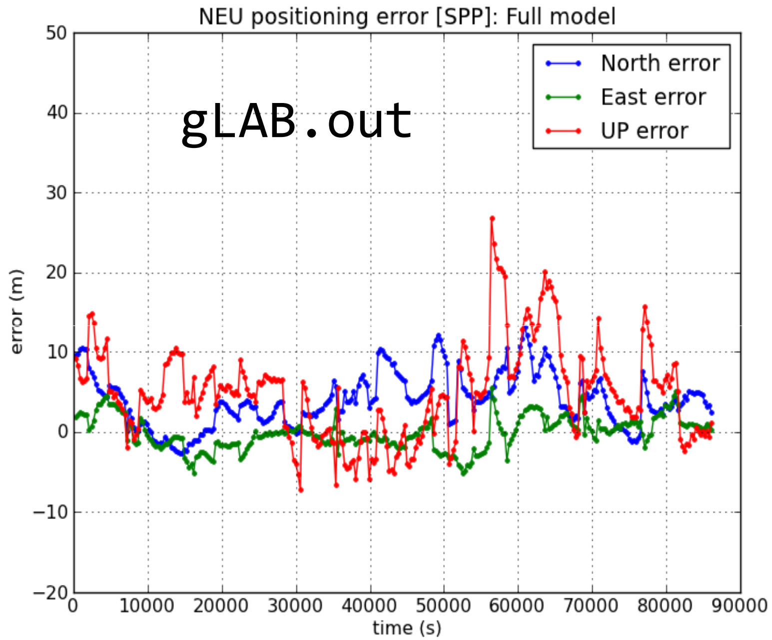
Equivalent command line sentence:

```
gLAB_linux -input:cfg gLAB_p1_Full.cfg
           -input:obs ramo1250.00o
           -input:nav brdc1250.00n
```

Example 1:
Iono effects on single freq. Positioning



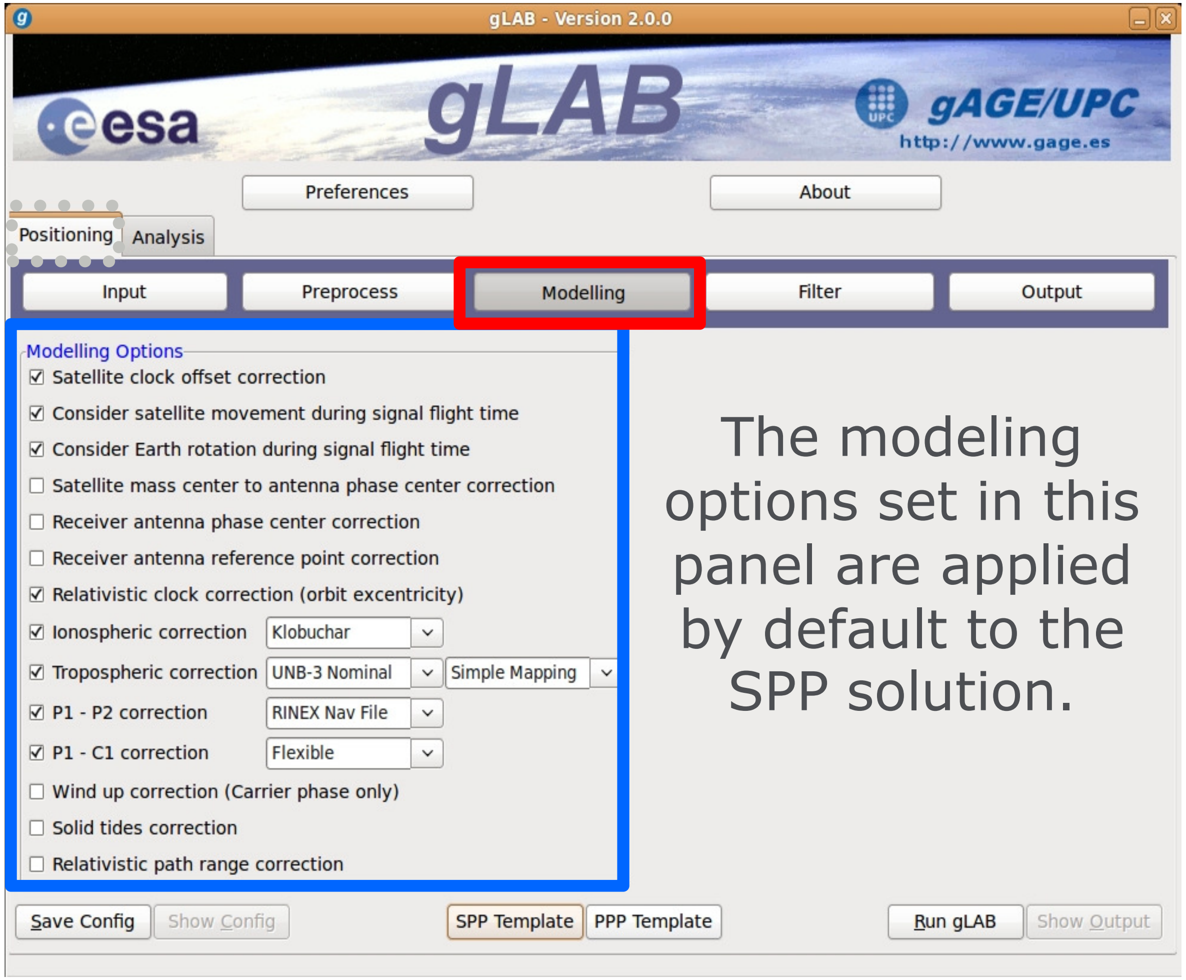
FULL SPP model



Equivalent command line sentence:

```
graph.py -f gLAB.out -x4 -y18 -s.- -c '($1=="OUTPUT")' -l "North error"
         -f gLAB.out -x4 -y19 -s.- -c '($1=="OUTPUT")' -l "East error"
         -f gLAB.out -x4 -y20 -s.- -c '($1=="OUTPUT")' -l "UP error“
                    --yn -20 --yx 50 --xl "time (s)" --yl "error (m)"
                   -t "NEU positioning error [SPP]: Full model"
```

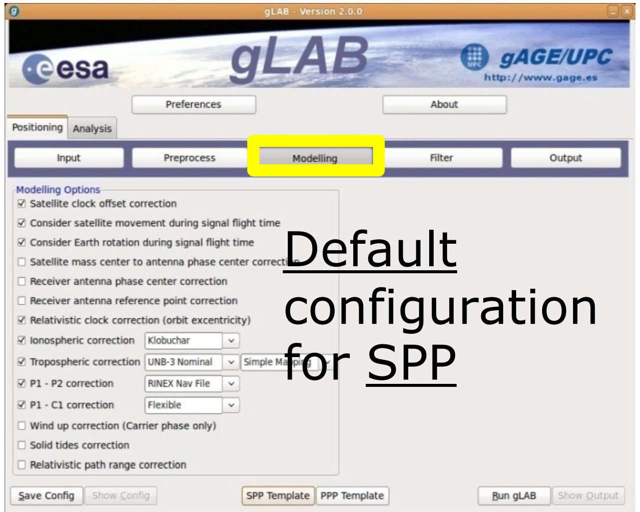
Example 1. gLAB Modeling panel



The different model terms can be analyzed with gLAB:

- Using the previous data file, the impact of neglecting the ionospheric correction is evaluated in the Range and Position domains.
- This is a baseline example of this analysis procedure. The same scheme must be applied for all model terms (troposphere, relativistic correction...). A full analysis of the different model components can be found in [R-2].)

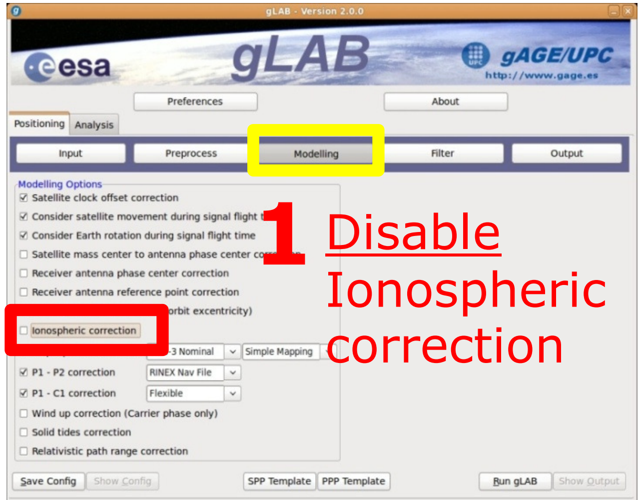
Example 1. Model component analysis: Ionosphere



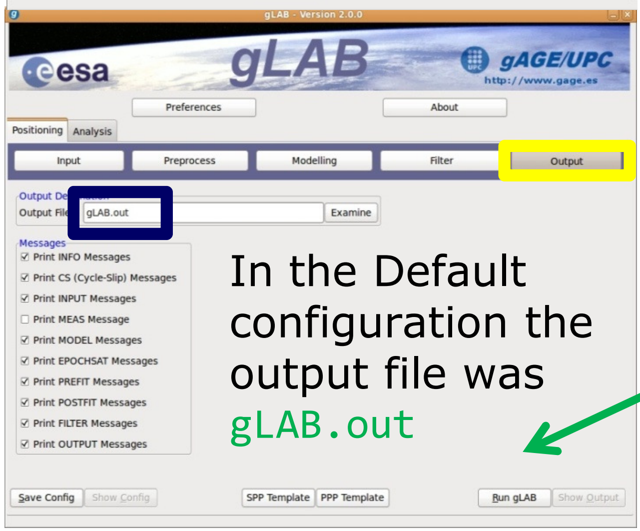
Default configuration for SPP

The procedure explained here is applicable for all model terms: iono, tropo…

1. In Modeling panel, disable the model component to analyze.
(in this example: disable Ionospheric correction)



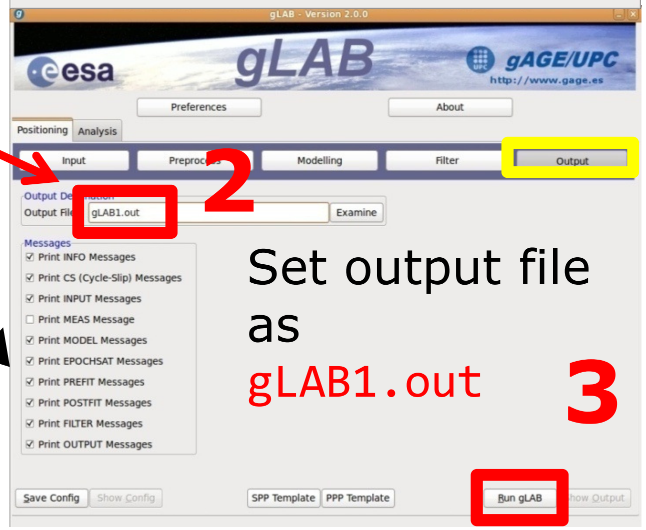
1 Disable Ionospheric correction



In the Default configuration the output file was gLAB.out

2. Save as **gLAB1.out** the associated output file.

Notice that the **gLAB.out** file contains the processing results with the **FULL model**, as was set in the default configuration.



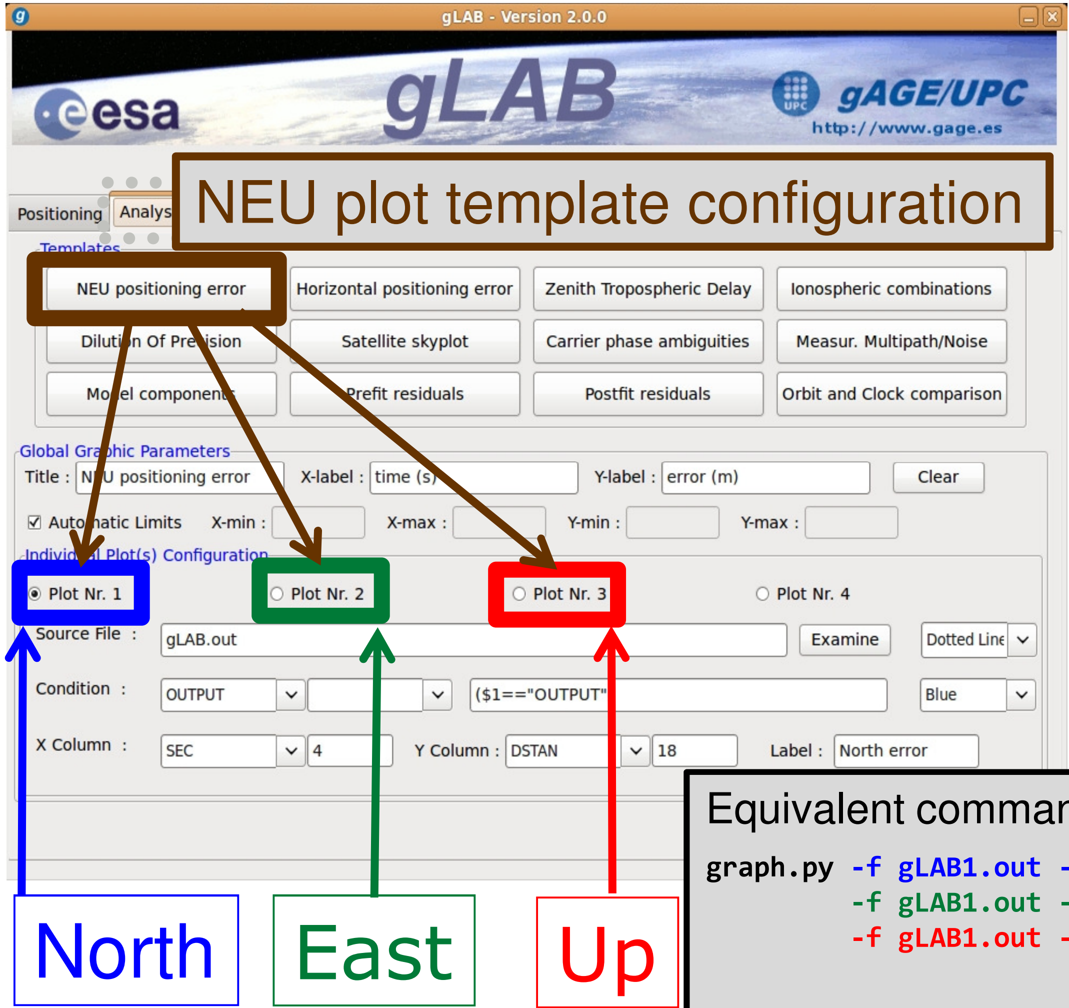
2

Set output file as gLAB1.out 3

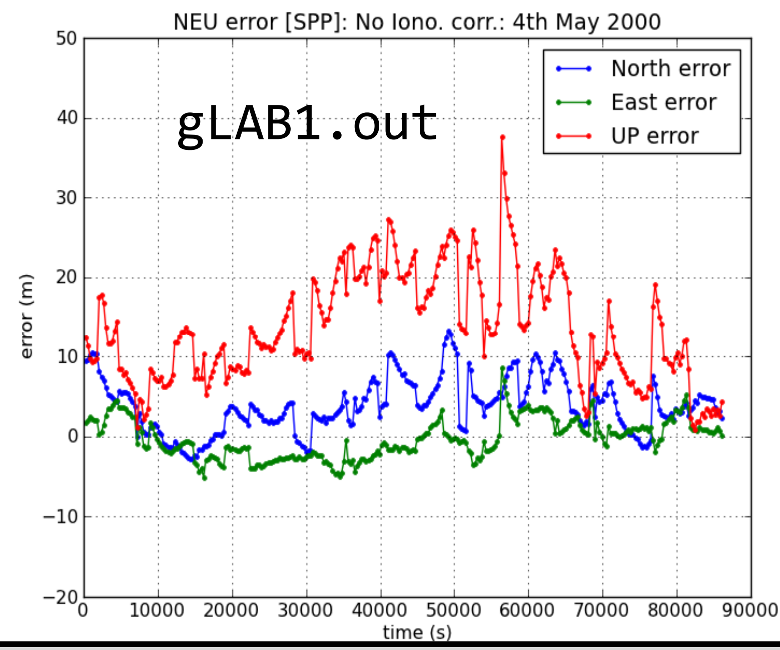
Equivalent command line sentence:

```
gLAB_linux -input:cfg gLAB_p1_NoIono.cfg
           -input:obs ramo1250.00o -input:nav brdc1250.00n
```

Example 1. NEU Position Error plot from gLAB1.out



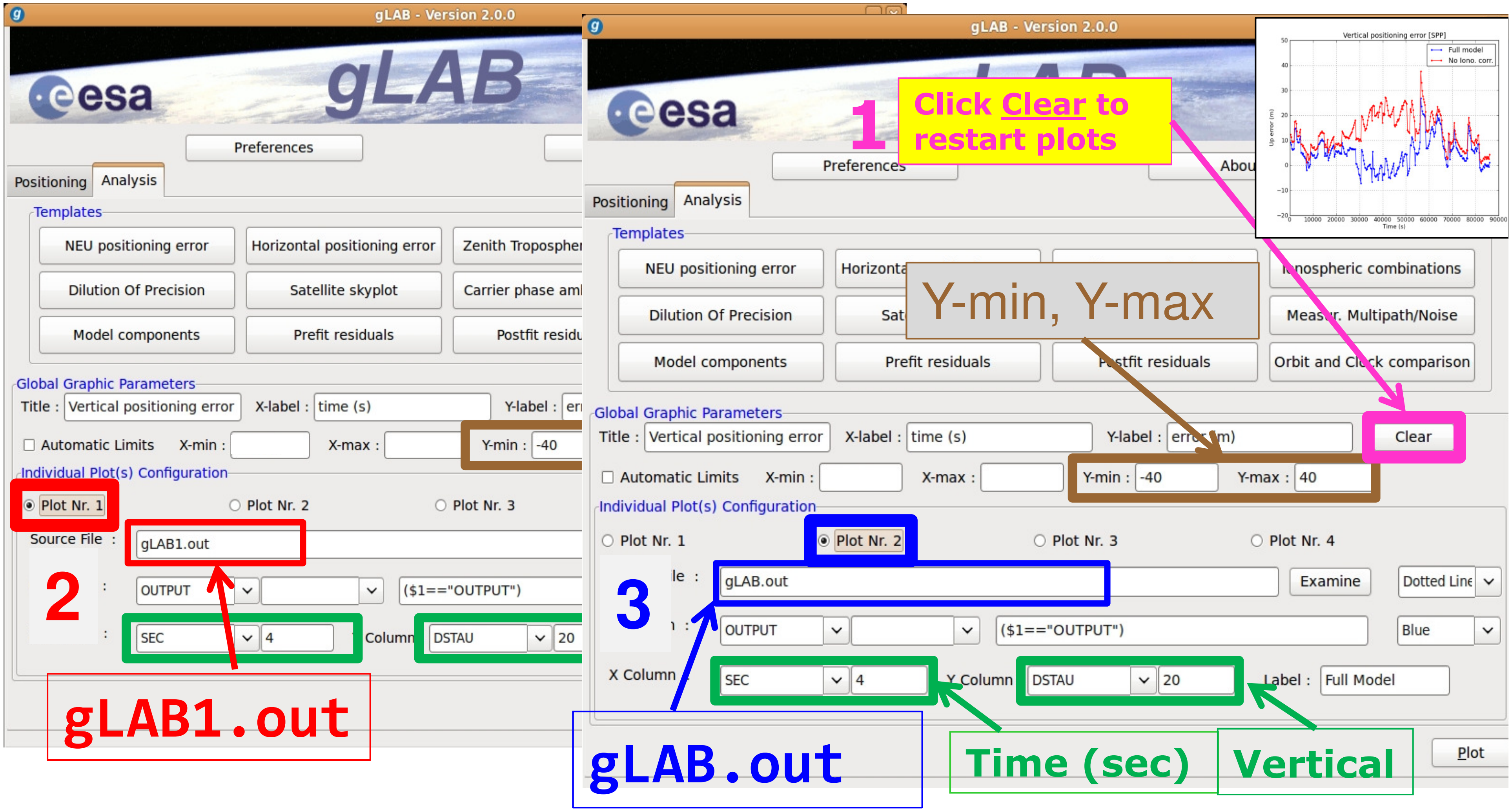
No Iono. correction



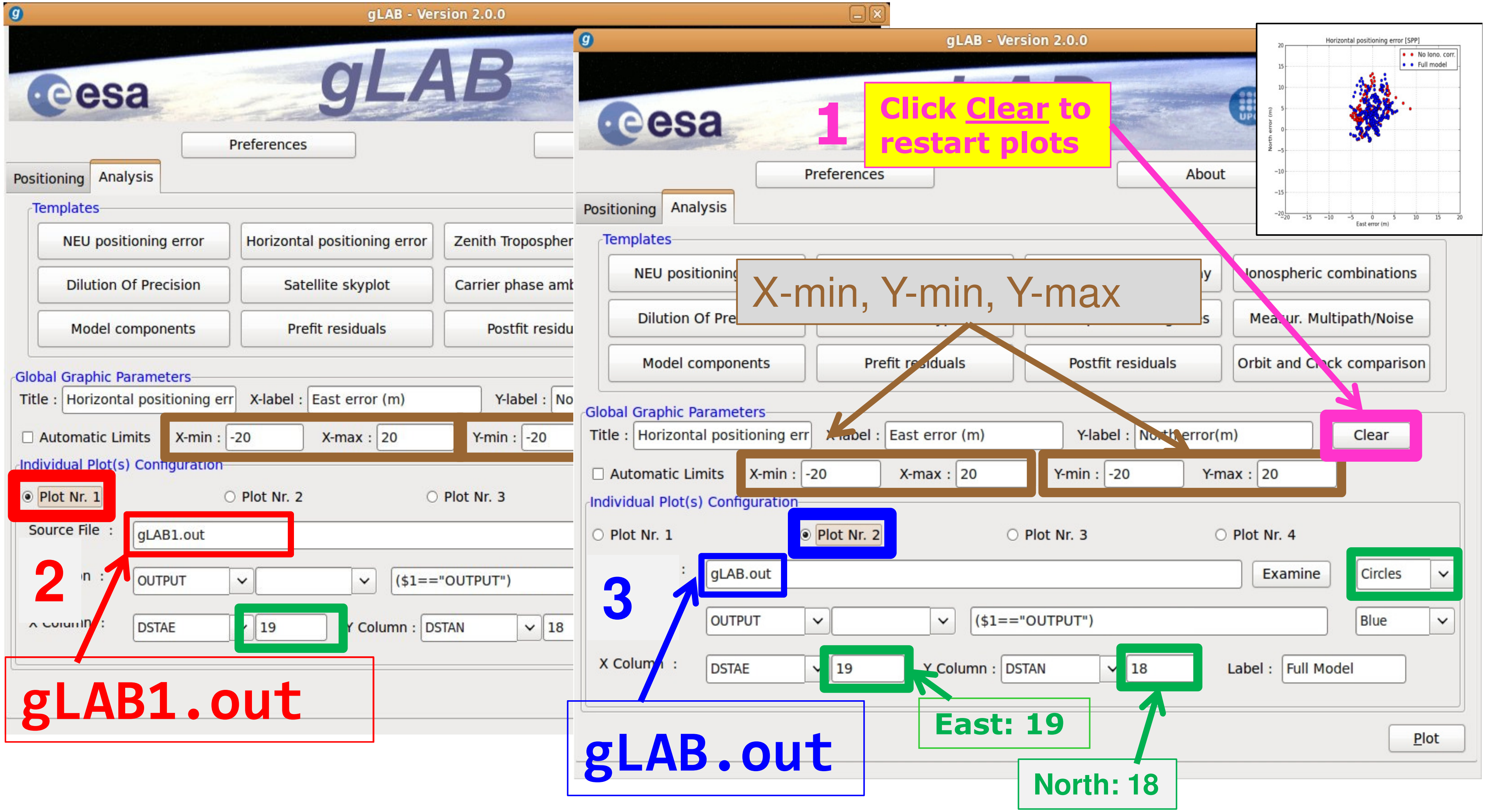
Equivalent command line sentence:

```
graph.py -f gLAB1.out -x4 -y18 -s.- -c '($1=="OUTPUT")' -l "North error"
         -f gLAB1.out -x4 -y19 -s.- -c '($1=="OUTPUT")' -l "East error"
         -f gLAB1.out -x4 -y20 -s.- -c '($1=="OUTPUT")' -l "UP error“
                       --yn -20 --yx 50 --xl "time (s)" --yl "error (m)"
                       -t "NEU positioning error [SPP]: No Iono. Corr."
```

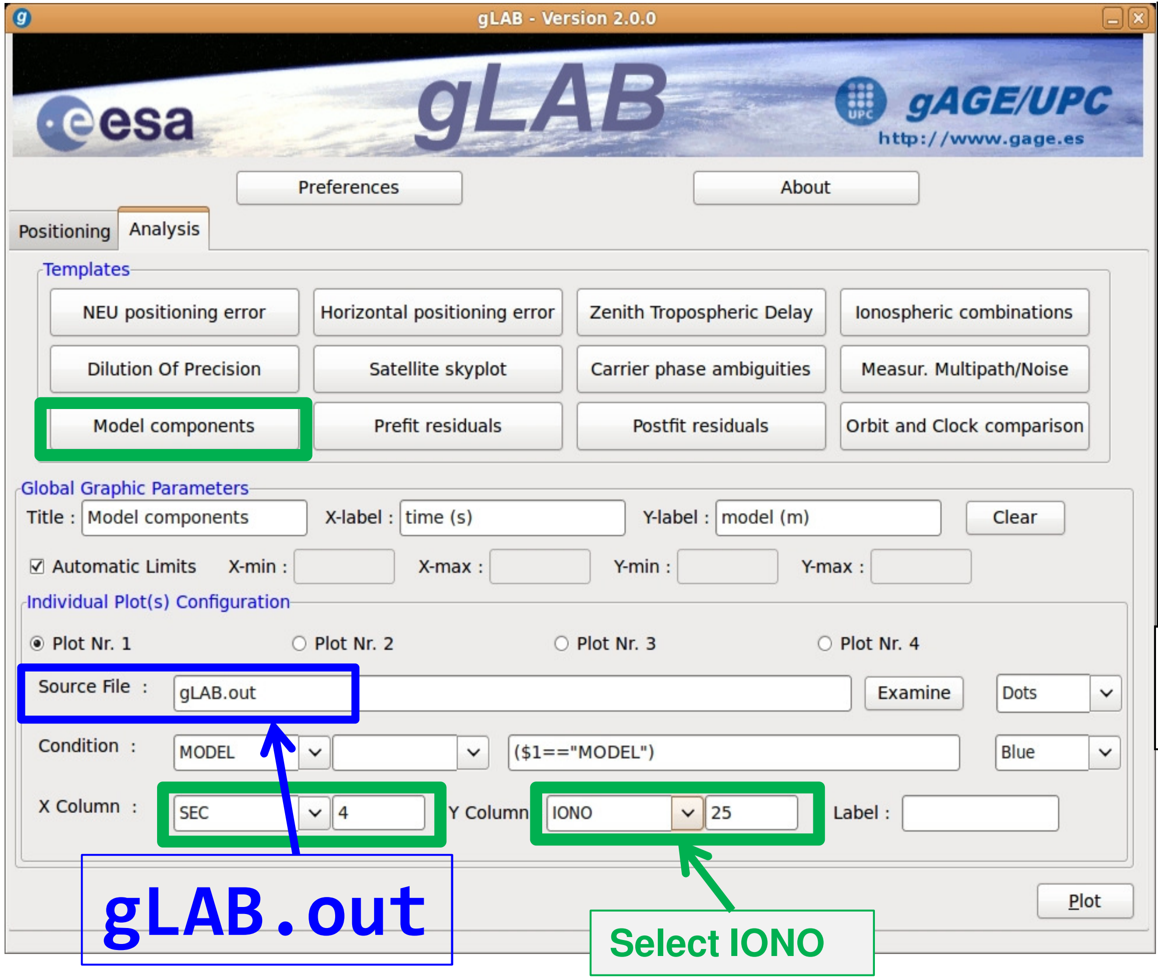
Example 1. VPE plot from gLAB.out,gLAB1.out

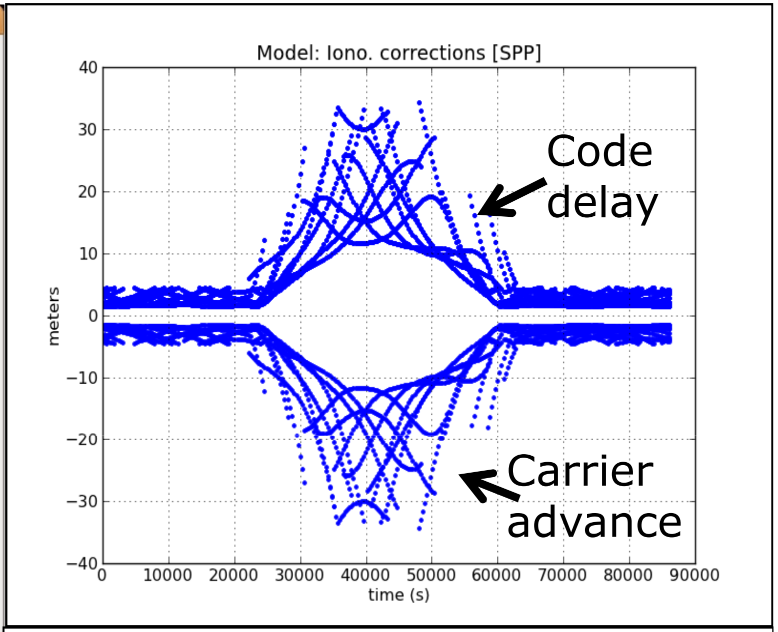


Example 1. HPE plot: gLAB.out, gLAB1.out



Example 1. Klobuchar iono. corr. plot: gLAB.out

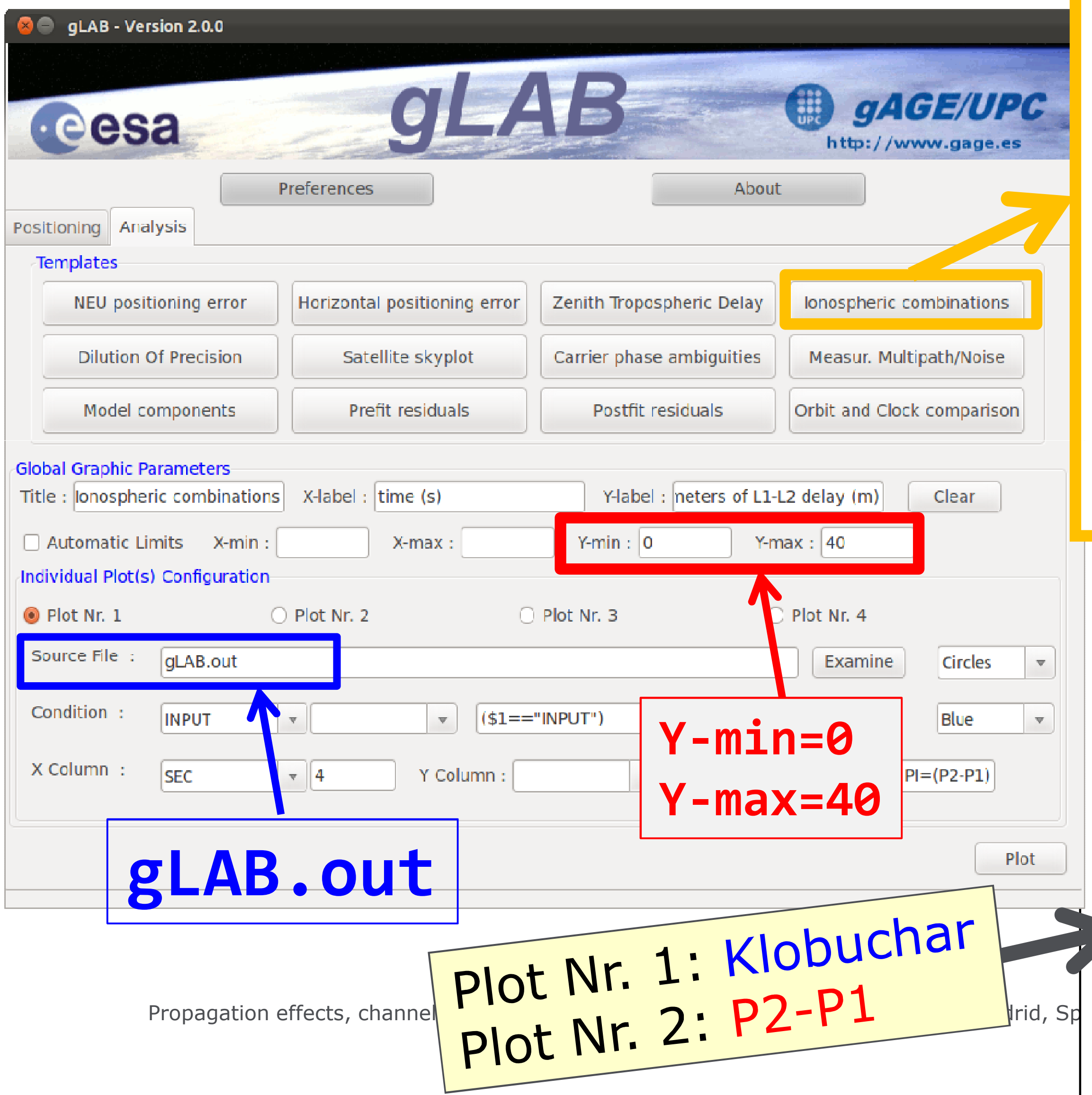


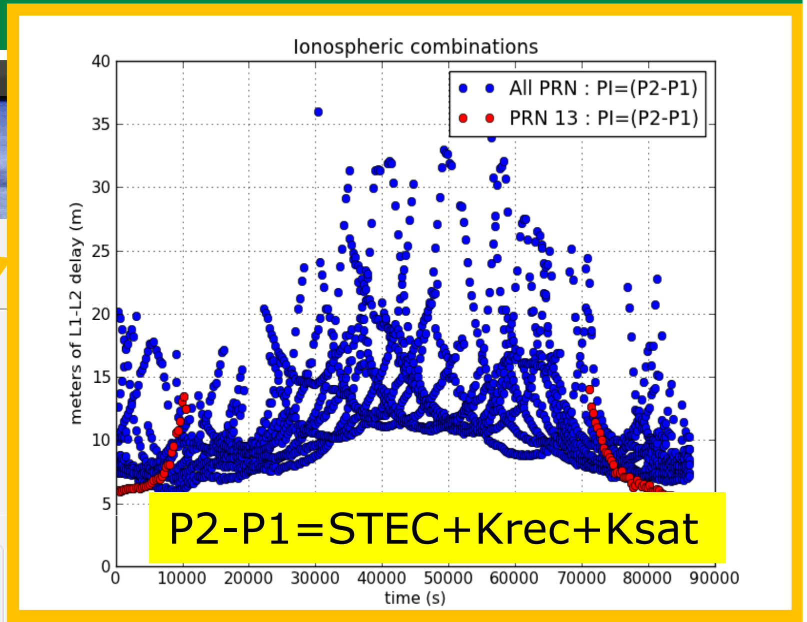


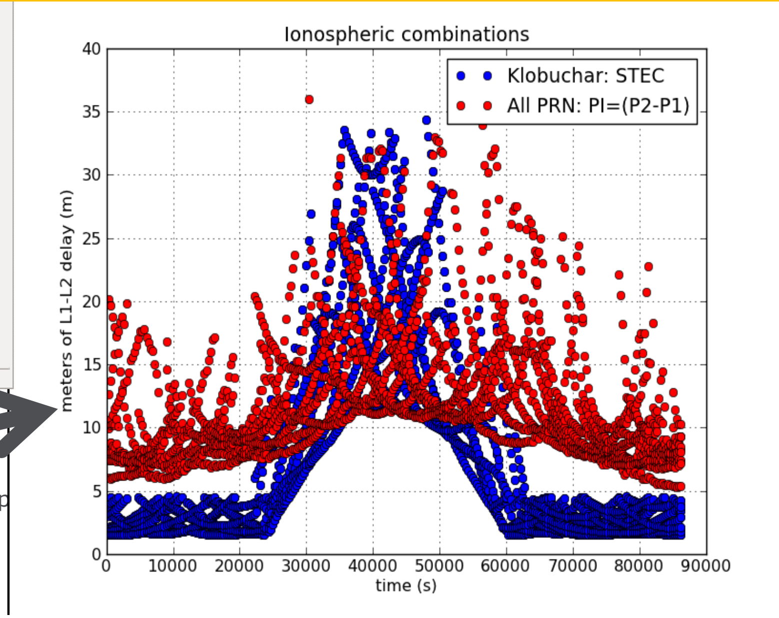
Ionosphere delays code and advances carrier measurements.

Note: Use the **gLAB.out** file. In **gLAB1.out** file this model component was switched off.

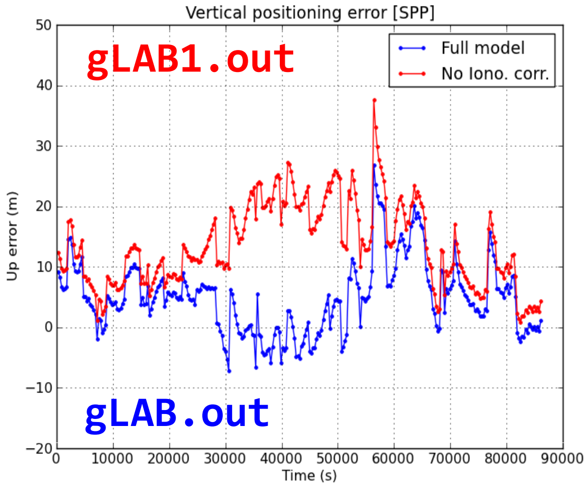
Example 1. Measur. (P2-P1) v.s. Model (Klobuchar)

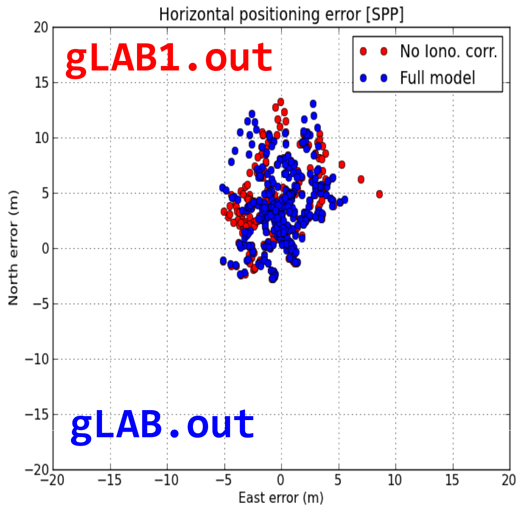


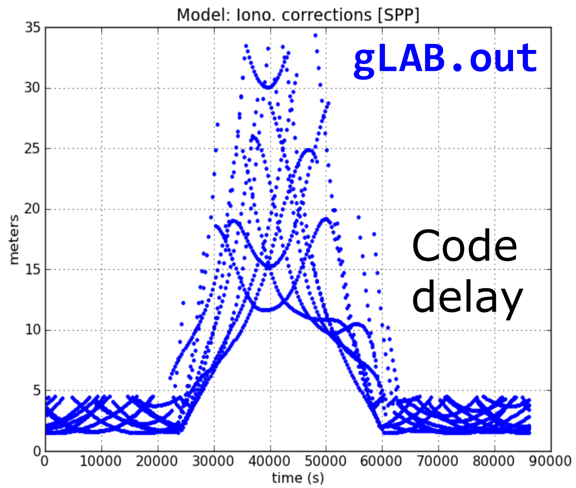




Example 1. Summary: Klobuchar model perform.







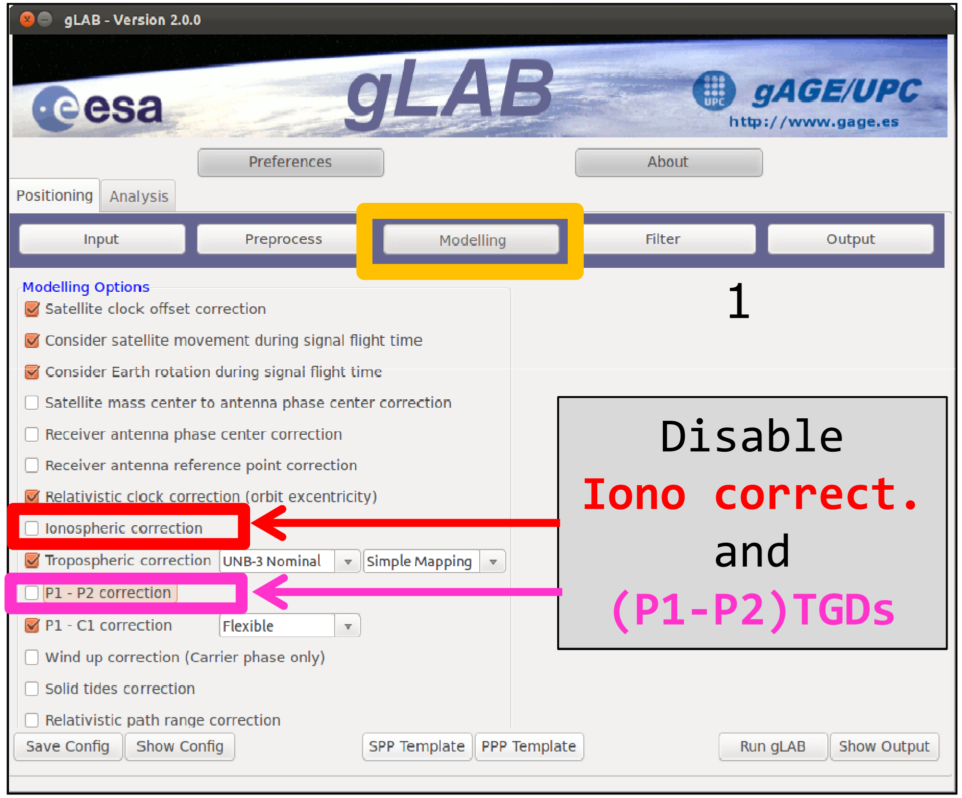
Ionospheric correction
(broadcast Klobuchar)

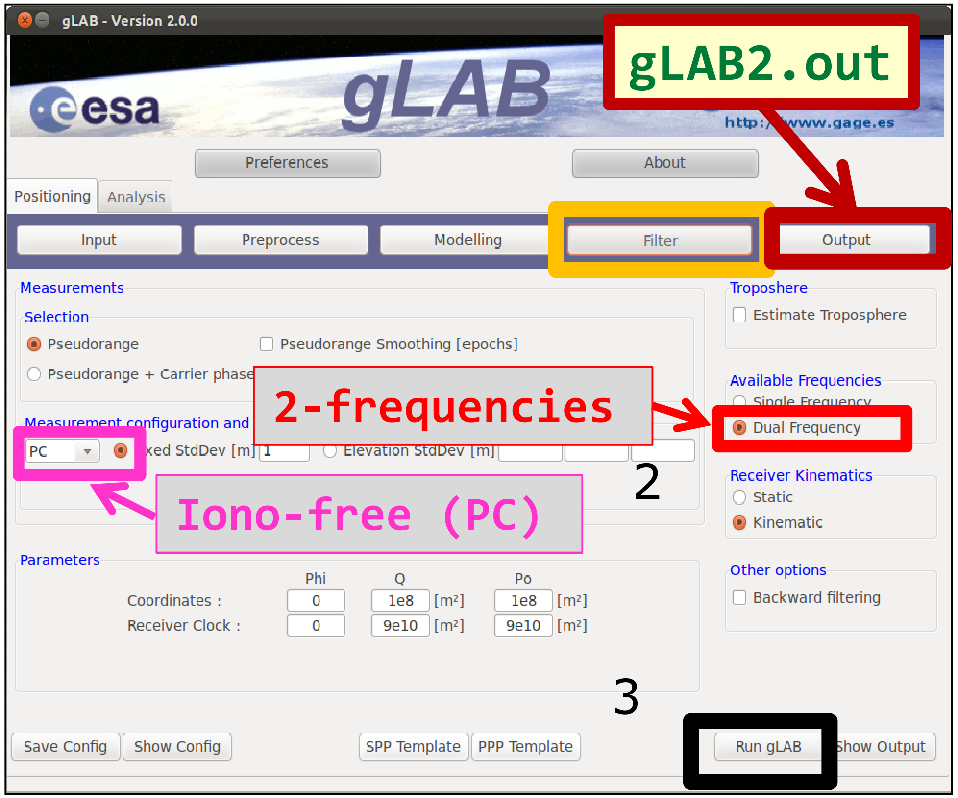
Ionospheric delays are larger at noon due to the higher illumination.

Large positioning errors (mainly in vertical) appear when neglecting ionospheric corrections.

Example 1. 2-frequency Ionosphere-free solution

From previous configuration set following options



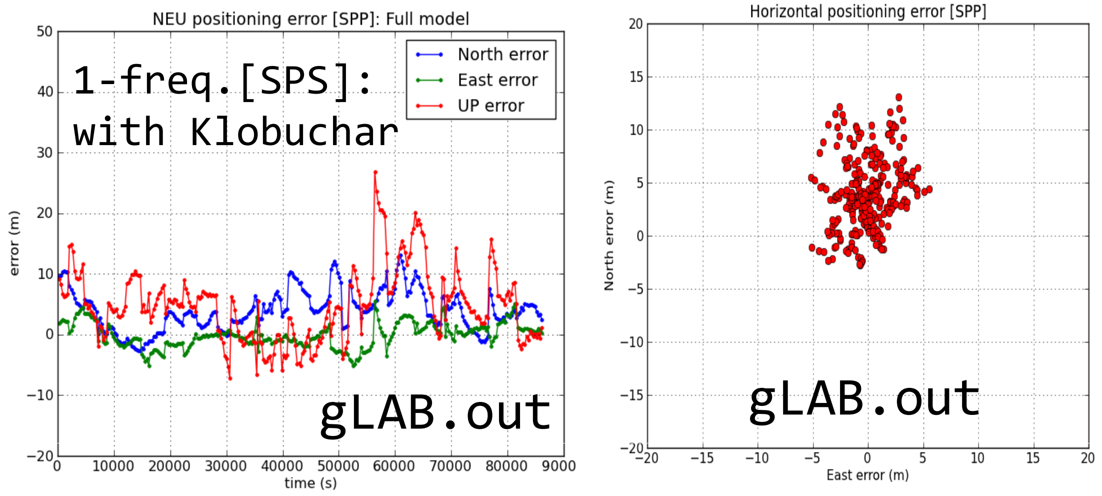


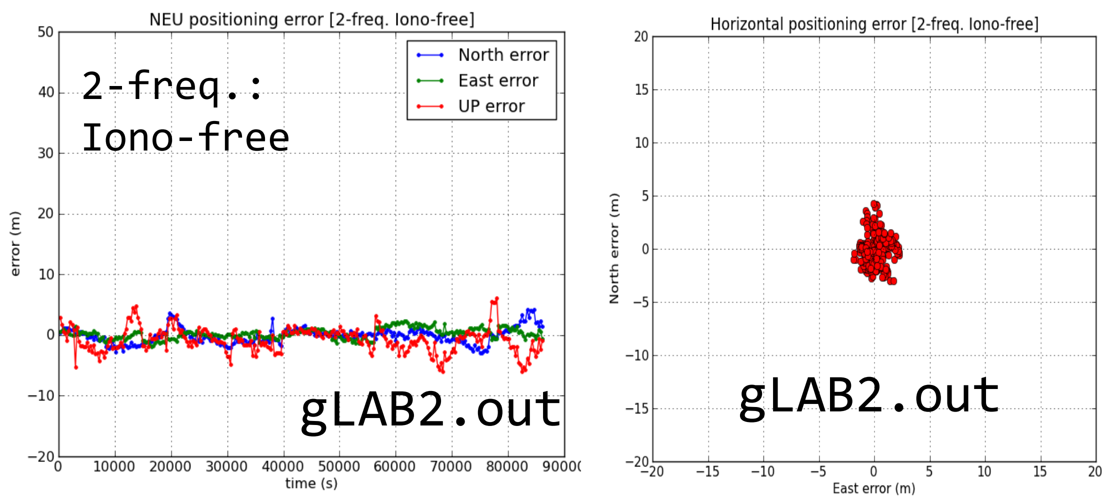
After running gLAB, plot results as in previous cases

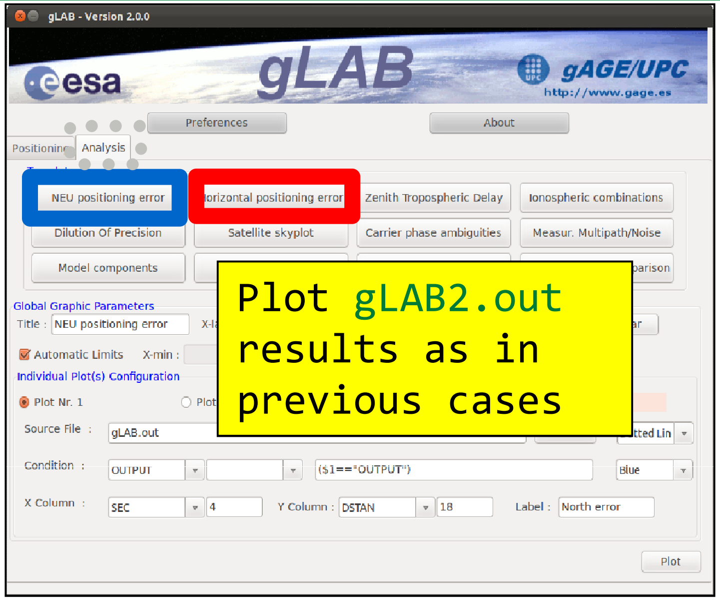
Equivalent command line sentence:

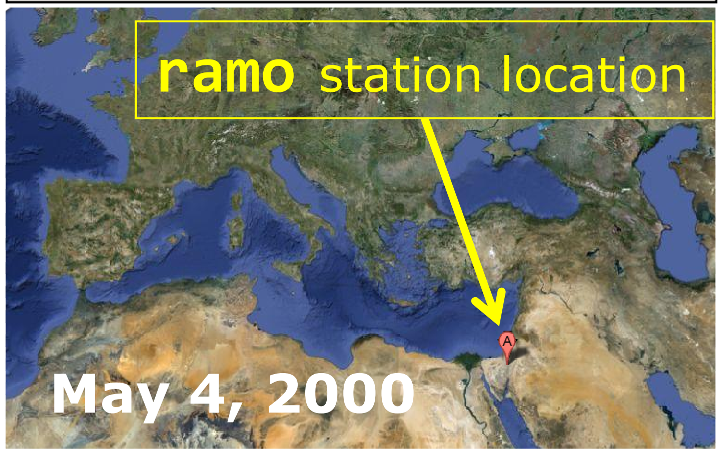
```
gLAB_linux -input:cfg gLAB_pc_IFree.cfg
           -input:obs ramo1250.00o -input:nav brdc1250.00n
```

Example 1. Single-frequency vs. Dual-frequency





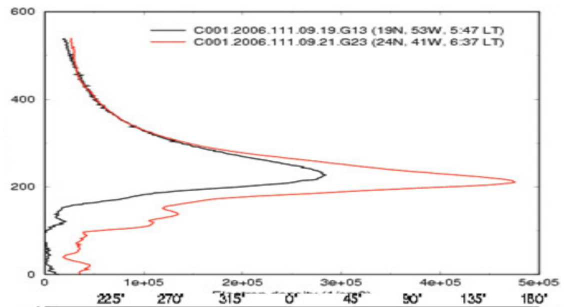


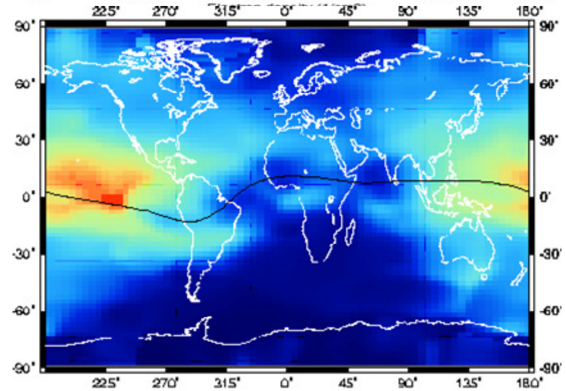


Example 1:
Iono effects on single freq. Positioning

Ionospheric delay

The ionosphere extends from about *60 km* over the Earth's surface until more than *2000 km*, with a sharp electron density maximum at around *350 km*. The ionospheric refraction depends, among other things, on the location, local time and solar cycle (*11 years*).





- First order (~99.9%) ionospheric delay $I1$ depends on the inverse of squared frequency:

$$I1 = \frac{40.3}{f^2} STEC$$

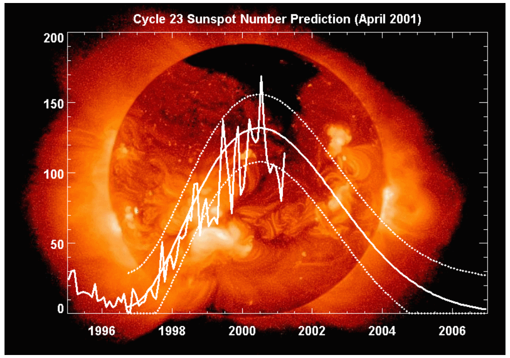
  where *STEC* is the number of electrons per area unit along ray path (STEC: Slant Total Electron Content).

$$STEC = \int N_e ds$$

- Two-frequency receivers can remove this error source (up to 99.9%) using ionosphere-free combination of pseudo-ranges (PC) or carriers (LC).

$$PC = \frac{f_1^2 P1 - f_2^2 P2}{f_1^2 - f_2^2}$$

- Single-frequency users can remove about a 50-70% of the ionospheric delay using the Klobuchar model, whose parameters are broadcast in the GPS navigation message.



Backup

Example 1:
Iono effects on single freq. Positioning

Annex:
gLAB processing in command line

Example 1: gLAB processing in command line

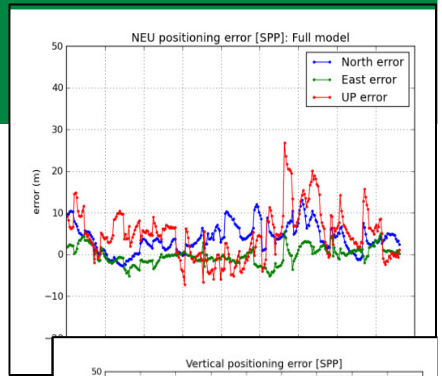
Execute <u>in a single line</u>: (<u>gnuplot</u> can also be used)

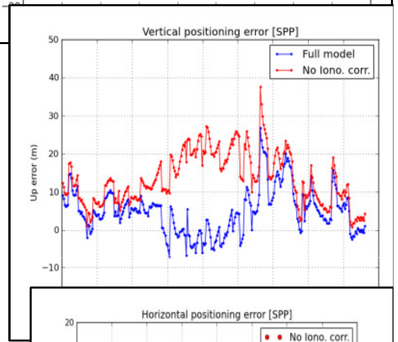
```
graph.py -f gLAB.out -x4 -y18 -s.- -c '($1=="OUTPUT")' -l "North error"
         -f gLAB.out -x4 -y19 -s.- -c '($1=="OUTPUT")' -l "East error"
         -f gLAB.out -x4 -y20 -s.- -c '($1=="OUTPUT")' -l "UP error"
                       --yn -20 --yx 50 --xl "time (s)" --yl "error (m)"
                      -t "NEU positioning error [SPP]: Full model"
```

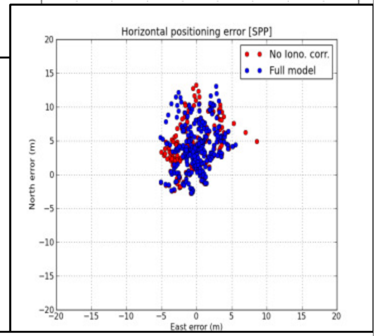
```
graph.py -f gLAB.out  -x4 -y20 -s.- -c '($1=="OUTPUT")' -l "Full model"
         -f gLAB1.out -x4 -y20 -s.- -c '($1=="OUTPUT")' -l "No Iono." --cl r
                      --yn -20 --yx 50 --xl "Time (s)" --yl "Up error (m)"
                      -t "Vertical positioning error [SPP]"
```

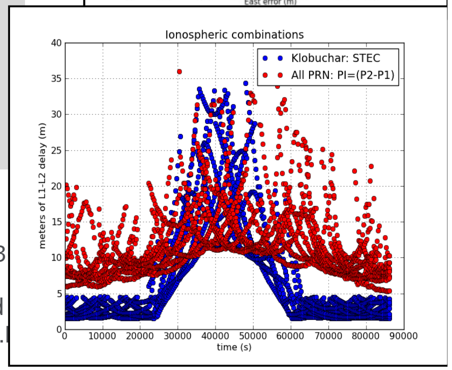
```
graph.py  -f gLAB1.out -x19 -y18 -so -c '($1=="OUTPUT")' -l "No Iono." --cl r
          -f gLAB.out  -x19 -y18 -so -c '($1=="OUTPUT")' -l "Full mod" --cl b
                      --xl "East error (m)" --yl "North error (m)"
                      --xn -20 --xx 20 --yn -20 --yx 20
                      -t "Horizontal pos. error [SPP]"
```

```
graph.py  -f gLAB.out -x4 -y25 -s. -c '($1=="MODEL")' -l "Klobuchar:STEC"
          -f gLAB.out -x4 -y'($10-$9)' -s. -c '($1=="INPUT")' --cl r -l "ALL PI"
                      --xl "time (s)" --yl "meters"
                      --yn -0 --yx 40  -t "Ionospheric Combination"
```



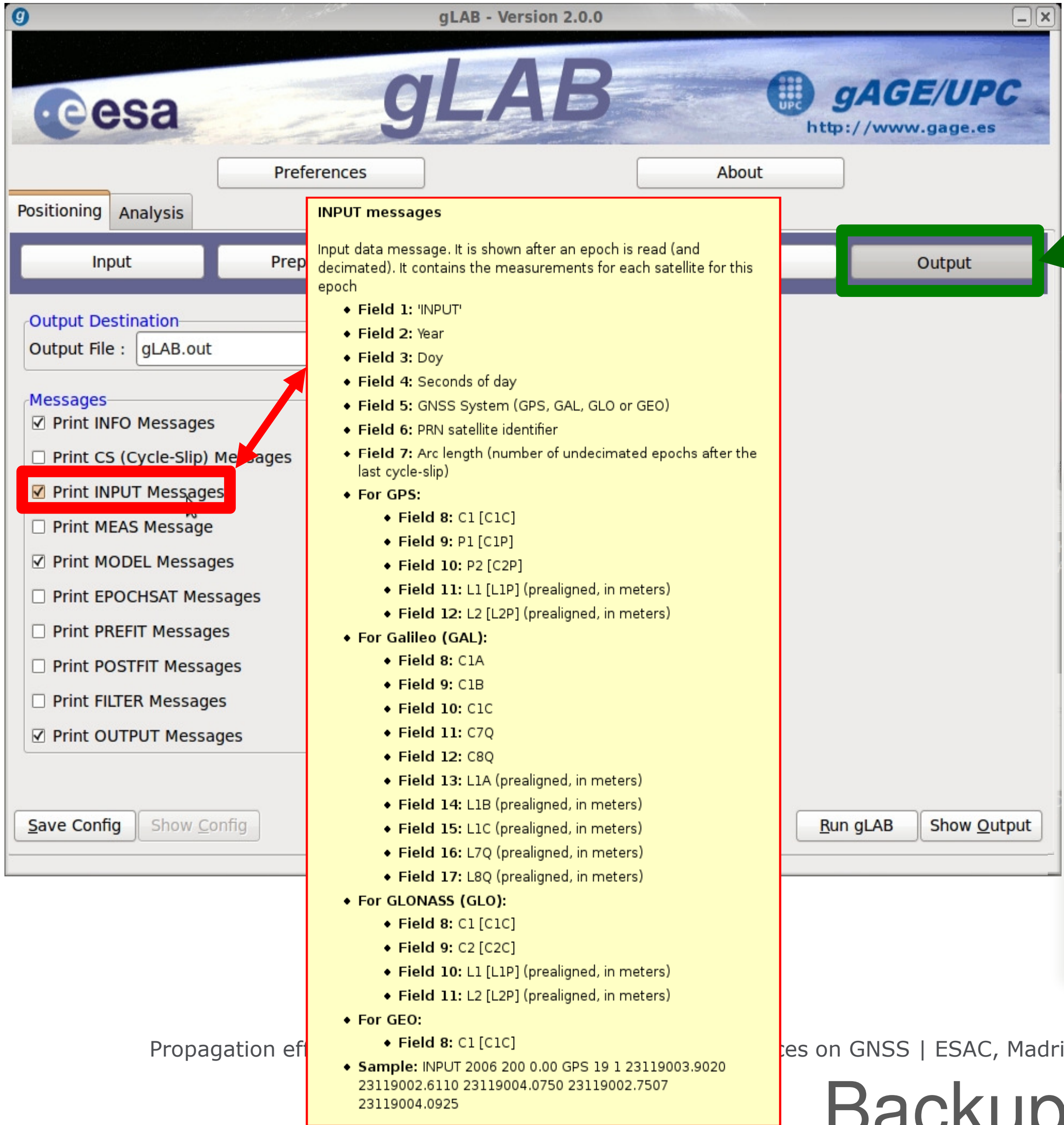






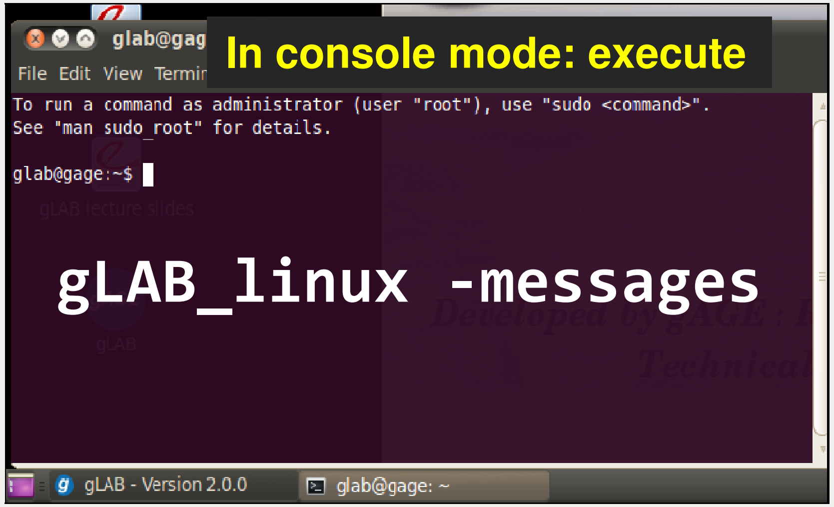
Tutorial associated
J. Sanz Subirana, J.

Example 1: gLAB processing in command line



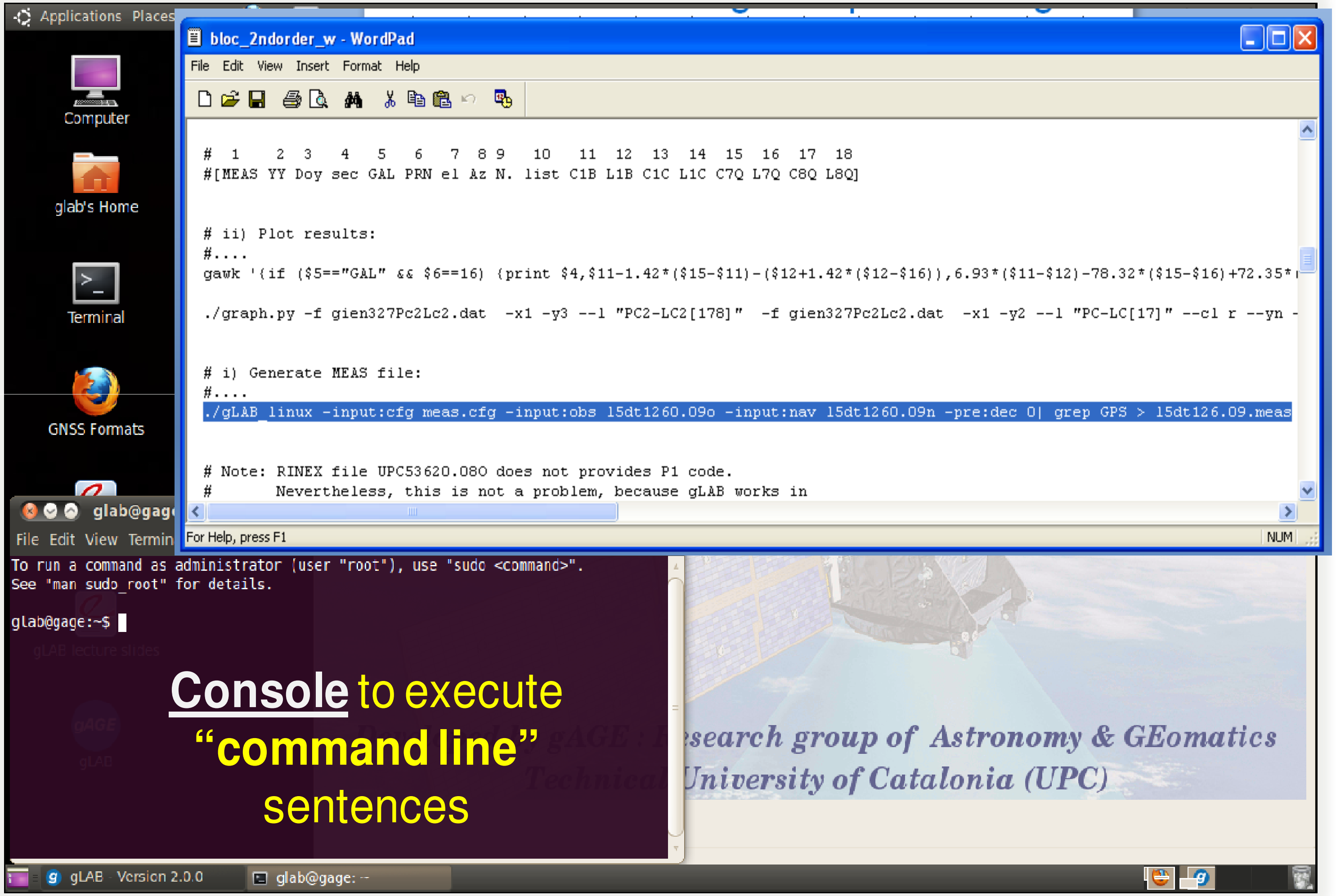
The different messages provided by gLAB and its content can be found in the [OUTPUT] section.

By placing the mouse on a given message name, a tooltip appears describing the different fields.



Backup

Example 1: gLAB processing in command line



A "notepad" with the command line sentence is provided to facilitate the sentence writing: just "copy" and "paste" from notepad to the working terminal.

Example 2: Ionospheric delay analysis

Example 2: Depict the ionospheric delays for the different satellites in view from station amc2

- This is a simple exercise aimed to illustrate how to use gLAB to easily analyze GNSS measurements and their combinations.

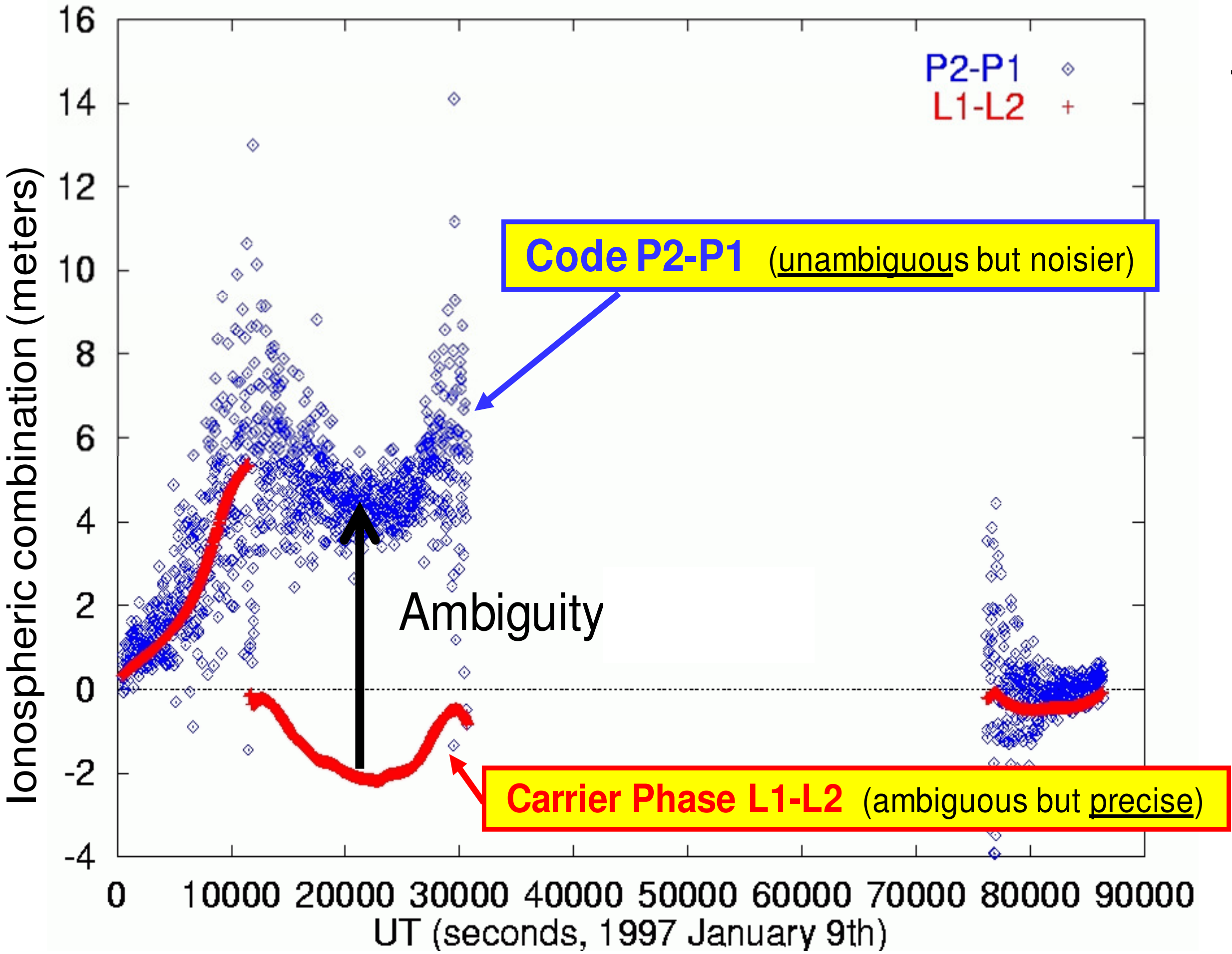
- gLAB will be used to read the RINEX measurements file and to generate a "text" with the measurements provided in a columnar format (more suitable to make plots).

- From "text" file, compute and plot the Ionospheric delay for a given satellite, by using code and carrier measurements at f1, f2:

$$P_I \equiv P_2 - P_1 = I + K_{21}$$
$$L_I \equiv L_1 - L_2 = I + Ambiguity$$

$$P_1 - L_1 = 2\tilde{\alpha}_1 I + ambiguity1$$
$$P_2 - L_2 = 2\tilde{\alpha}_2 I + ambiguity2$$

Example 2: Ionospheric delay analysis



The target is to generate this plot to depict the ionospheric delay from code & carrier data

$$P_I \equiv P_2 - P_1 = I + K_{21}$$
$$L_I \equiv L_1 - L_2 = I + Ambiguity$$

$$P_1 - L_1 = 2\tilde{\alpha}_1 I + ambiguity1$$

$$P_2 - L_2 = 2\tilde{\alpha}_2 I + ambiguity2$$

$$\tilde{\alpha}_1 = \frac{1}{\gamma_{21} - 1} = 1.546\ ;\ \ \tilde{\alpha}_2 = 1 + \tilde{\alpha}_1$$

$$\gamma_{21} = \left(f_1 / f_2\right)^2 = (154/120)^2$$

Example 2: GPS measurements content

Code measurements

$$P_1 = \rho + \tilde{\alpha}_1 (I + K_{21}) + \varepsilon_1$$
$$P_2 = \rho + \tilde{\alpha}_2 (I + K_{21}) + \varepsilon_2$$

Carrier measurements

$$L_1 = \rho - \tilde{\alpha}_1 I + B_1 + \varsigma_1$$
$$L_2 = \rho - \tilde{\alpha}_2 I + B_2 + \varsigma_2$$

$$\tilde{\alpha}_2 - \tilde{\alpha}_1 = 1$$
$$\tilde{\alpha}_1 = \frac{1}{\gamma_{21} - 1} = 1.546$$
$$\gamma_{21} = (f_1 / f_2)^2$$

ρ Refers to all non dispersive terms: geometric range, clocks, tropo. delay… (see [R-1]).

Ionospheric delay

$$I = \frac{40.3(f_1^2 - f_2^2)}{f_1^2 f_2^2} 10^{16} STEC \,; \quad (I \text{ is in } m \text{ of } L_1 - L_2 \text{ delay})$$

$$STEC = \int_{rec}^{sat} N_e dl \,, \quad (STEC \text{ is in } TECUs)$$

$$1\, TECU = 10^{16} e^- / m^2 = 0.10 m \text{ of } L1\text{-}L2 \text{ delay}$$

Interfrequency bias

$$K_{21} = K_{21,rec} - K_{21}^{sat}$$

As the satellite clocks are referred to the ionosphere-free combination of codes (P_C), the K_{21}^{sat} cancels in such combination.
Note: $TGD = -\tilde{\alpha}_1 K_{21}^{sat}$ is broadcast in GPS nav. Message.

$$P_C = \frac{f_1^2 P_1 - f_2^2 P_2}{f_1^2 - f_2^2}$$

Carrier ambiguities

$$B_i = \lambda_i N_i + b_i$$

N_i is an integer number. b_i is a real number (fractional part of ambiguity)

Backup

Example 2: Ionospheric delay analysis

1.- Read RINEX file with gLAB and generate a "measurements file" in a columnar format (the easiest to manipulate and plot content):

→ Using the configuration file **meas.cfg**, READ the RINEX and generate the MEAS file

```
gLAB_linux -input:cfg meas.cfg -input:obs coco0090.97o -input:nav brdc0090.97n > coco.meas
```

gLAB configuration file

```
-pre:dec 1
-print:none
-print:meas

--model:satphasecenter
--model:recphasecenter
--model:satclocks
--pre:cs:li
--pre:cs:bw
```

RINEX Measurement file

RINEX Navigation file

OUTPUT: measurement file in columnar format

```
[Id YY Doy sec GPS PRN el Az N. list C1C L1C C1P L1P C2P L2P]
  1  2   3   4   5   6  7  8  9  10  11  xx  xx  14  15 16 ]
```

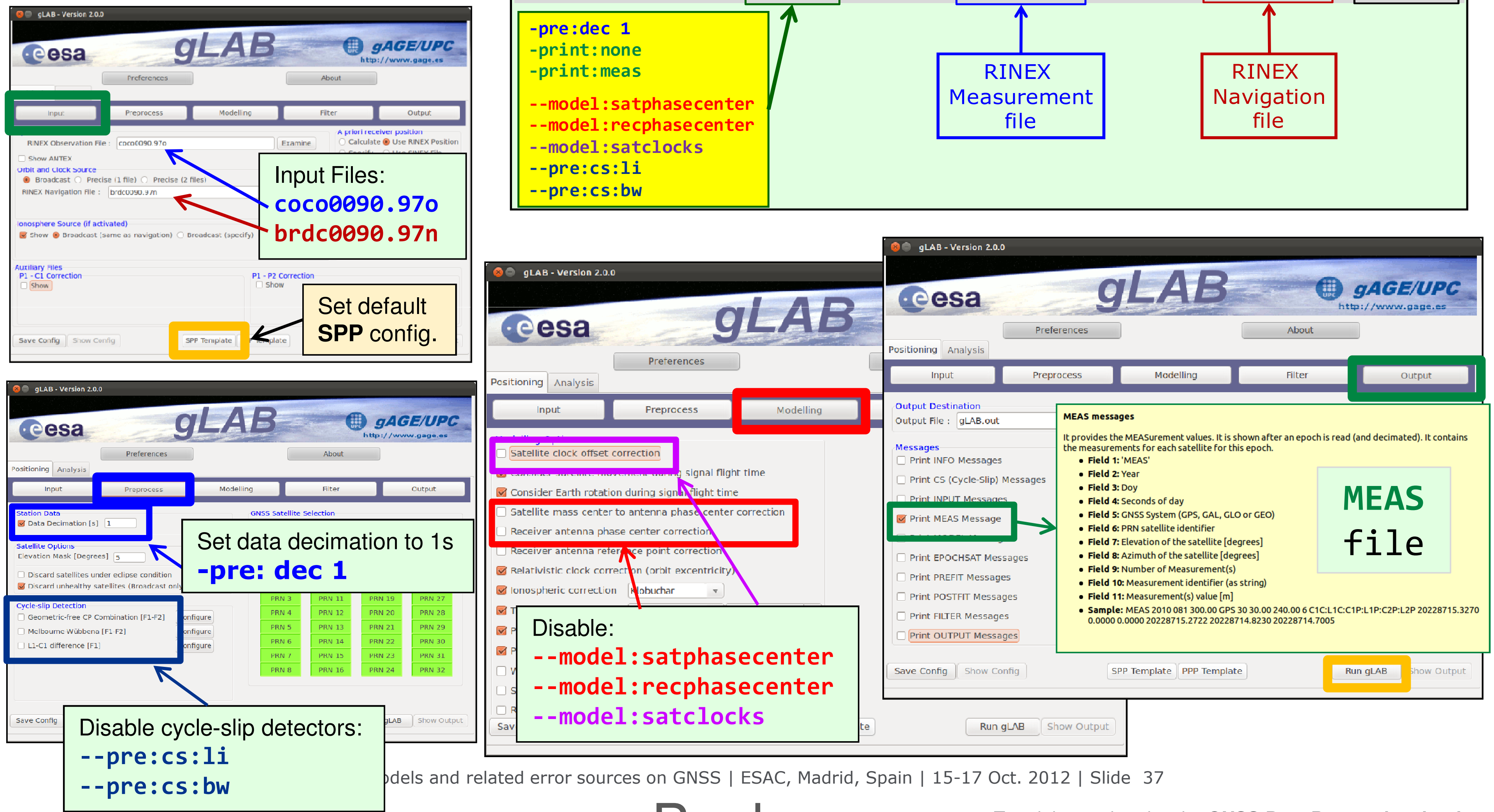
Example 1: gLAB processing in command line

```
gLAB_linux -input:cfg meas.cfg -input:obs coco0090.97o -input:nav brdc0090.97n > coco.meas
```

```
-pre:dec 1
-print:none
-print:meas
--model:satphasecenter
--model:recphasecenter
--model:satclocks
--pre:cs:li
--pre:cs:bw
```

RINEX Measurement file

RINEX Navigation file



Backup

Tutorial associated to the **GNSS Data Processing book**
J. Sanz Subirana, J.M. Juan Zornoza, M. Hernández-Pajares

Example 2: Ionospheric delay analysis

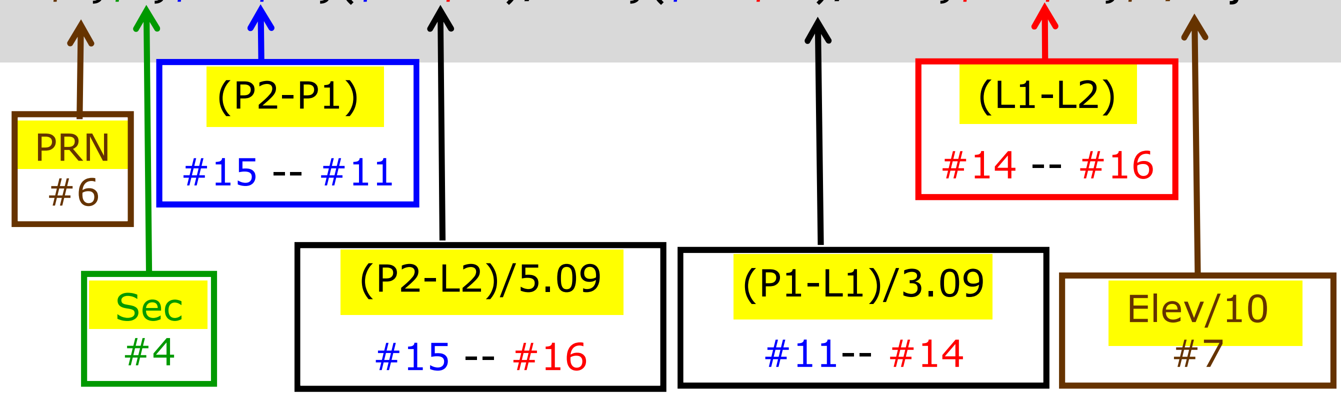
2.- Manipulate the file with the easy and powerful **awk** (or gawk) programming language (to compute the combinations of measurements):

→ From **coco.meas** file:

										P1			L1	P2	L2
[Id	YY	Doy	sec	GPS	PRN	el	Az	N.	list	C1C	L1C	C1P	L1P	C2P	L2P]
1	2	3	4	5	6	7	8	9	10	11	xx	xx	14	15	16

Compute different ionospheric combination of codes and carriers, and generate the obs.txt file containing the fields: [PRN,sec, P2-P1, (P2-L2)/5.09, (P1-L1)/3.09, L1-L2, Elev/10]

```
gawk '{print $6,$4,$15-$11,($15-$16)/5.09,($11-$14)/3.09,$14-$16,$7/10}' coco.meas > obs.txt
```



Example 2: Ionospheric delay analysis

3.-Plot results with "**graph.py**" (you can use the **gnuplot** as well)

→From **obs.txt** file:

```
[PRN, sec, P2-P1, (P2-L2)/5.09, (P1-L1)/3.09, L1-L2, Elev/10]
  1    2     3         4             5          6       7
```

Show in the same plot the following ionopheric delays for satellite **PRN01**:
P2-P1, (P2-L2)/5.09, (P1-L1)/3.09, L1-L2, Elev./10

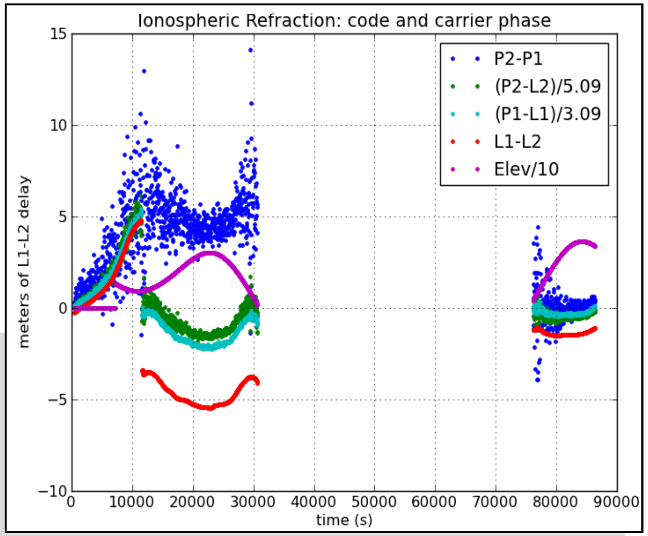
Condition: Select **PRN01** (from 1-st field)

Fields to plot: **#2** (x-axis) versus **#3** (y-axis)

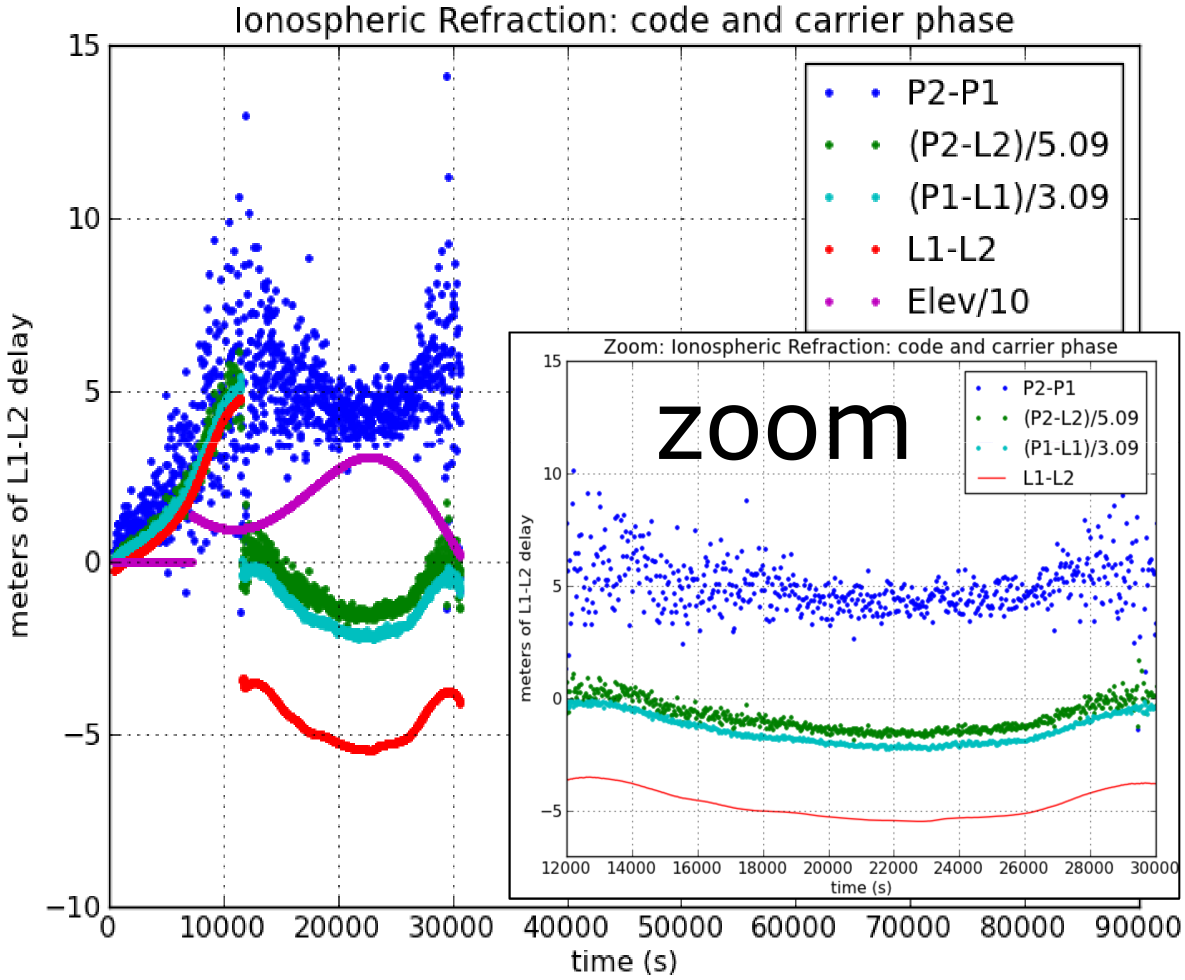
Label: P2-P1

File to plot

```
graph.py -f obs.txt -c'($1==01)' -x2 -y3 -l "P2-P1"
         -f obs.txt -c'($1==01)' -x2 -y4 -l "(P2-L2)/5.09"
         -f obs.txt -c'($1==01)' -x2 -y5 -l "(P1-L1)/3.09"
         -f obs.txt -c'($1==01)' -x2 -y6 -l "L1-L2"
         -f obs.txt -c'($1==01)' -x2 -y7 -l "Elev/10"
         --yn -10 --yx 15 --xl "time (s)" --yl "meters of L1-L2 delay"
```



Example 2: Ionospheric delay analysis



$$P_I \equiv P_2 - P_1 = I + K_{21}$$
$$L_I \equiv L_1 - L_2 = I + Ambiguity$$

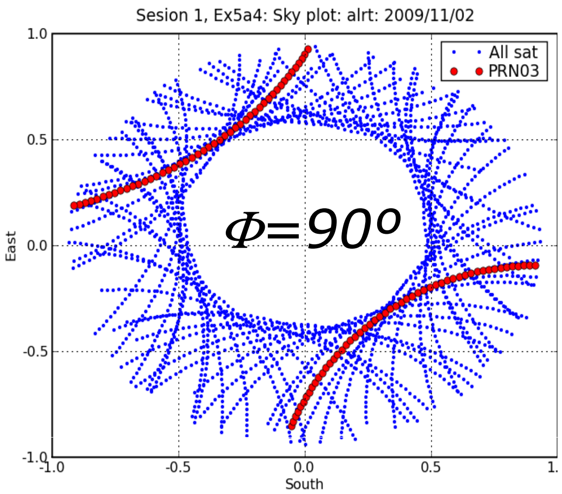
$$P_1 - L_1 = 2\tilde{\alpha}_1 I + ambiguity1$$

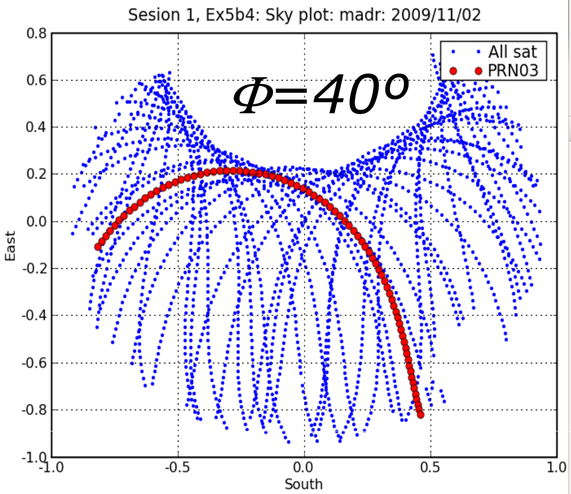
$$P_2 - L_2 = 2\tilde{\alpha}_2 I + ambiguity2$$

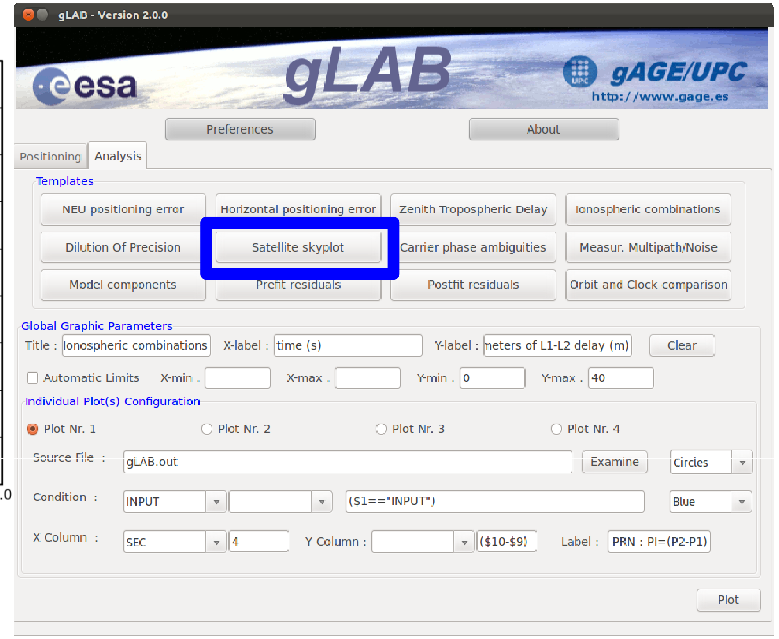
$$\tilde{\alpha}_1 = \frac{1}{\gamma_{21} - 1} = 1.546\ ;\ \tilde{\alpha}_2 = 1 + \tilde{\alpha}_1$$

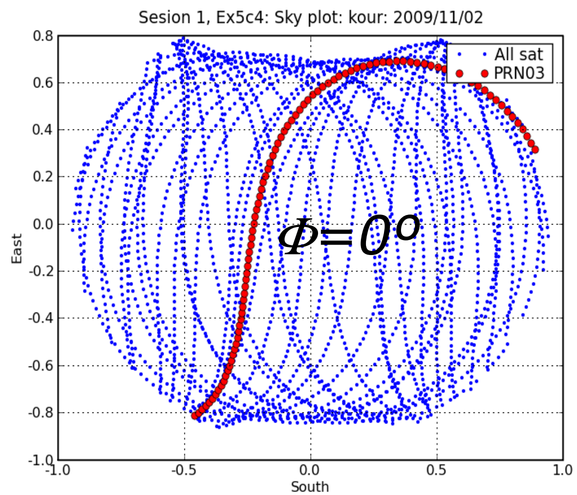
$$\gamma_{21} = \left(f_1 / f_2\right)^2 = (154/120)^2$$

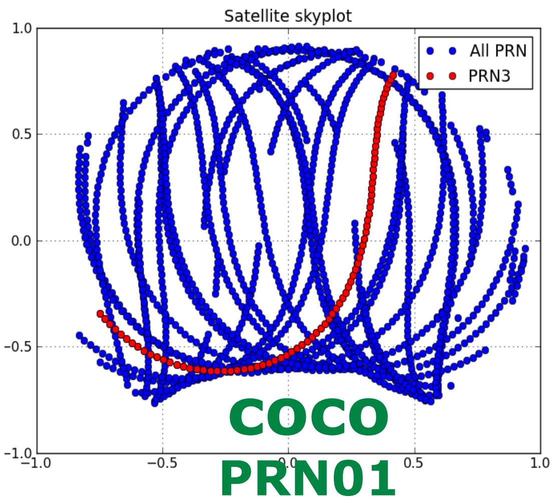
Example 2: Sky plots











Sky plots at different latitudes

Example 3: Zenith Troposphere Delay estimation

PPP Template: Static positioning with dual freq. code & carrier (iono-free combination PC,LC) + post-processed precise orbits & clocks.

1. Select the **PPP Template**
2. Upload data files:
   - Measurement : `roap1810.09o`
   - ANTEX: `igs05_1525.atx`
   - Orbits & clocks: `igs15382.sp3`
   - SINEX: `igs09P1538.snx`
3. RUN gLAB

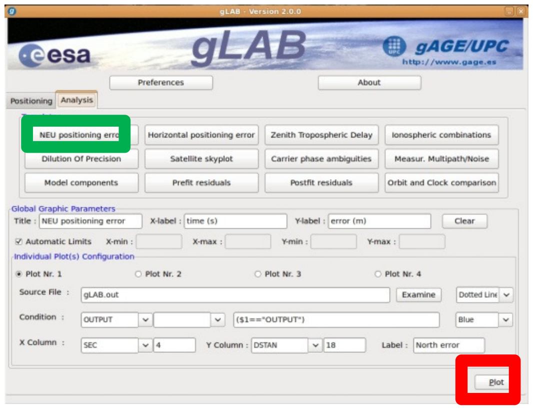
⇩

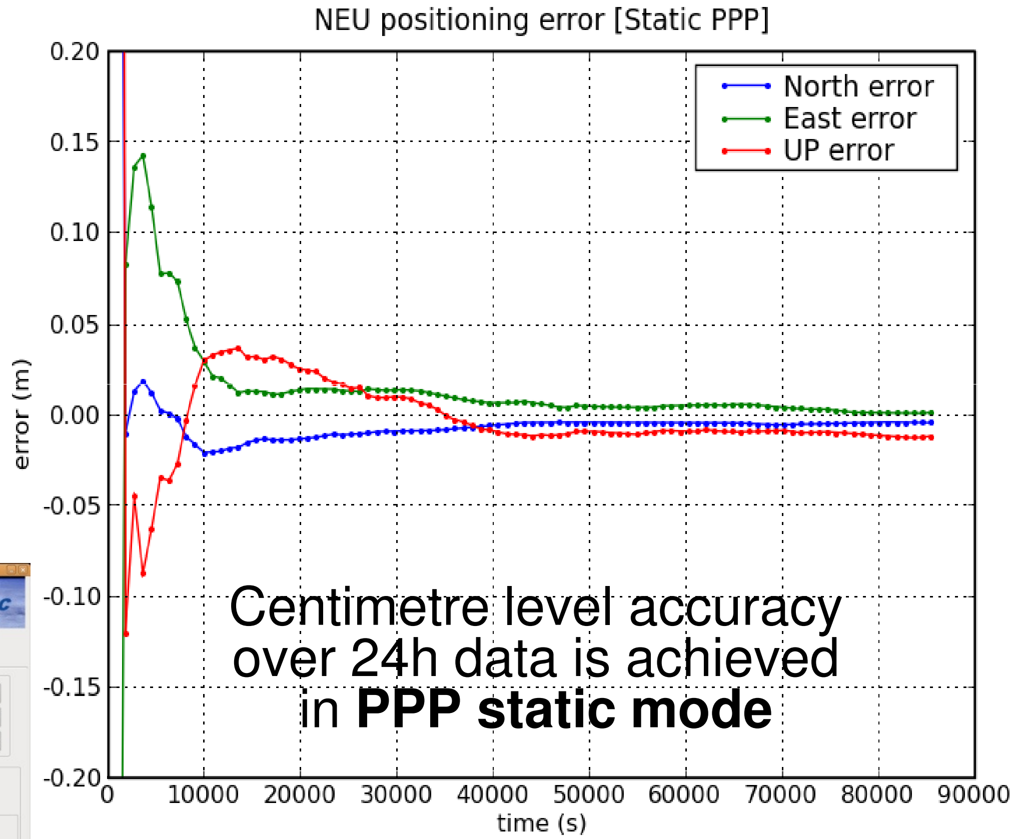
Default output file:
gLAB.out

Example 3: Zenith Troposphere Delay estimation

Plotting Results

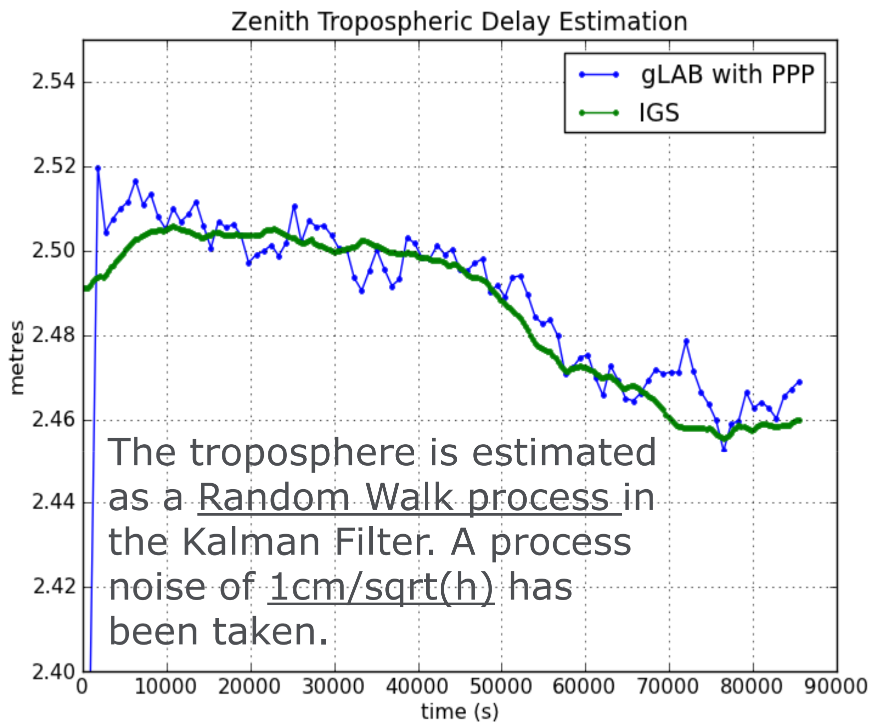
- Coordinates are taken as constants in nav. filter.
- Dual frequency Code and Carrier measurements.
- Precise orbits and clocks.
- Measurements modelling at the centimetre level.





Tutorial associated to the **GNSS Data Processing book**
J. Sanz Subirana, J.M. Juan Zornoza, M. Hernández-Pajares

Example 3: Zenith Troposphere Delay estimation



ftp://cddis.gsfc.nasa.gov/pub/gps/products/troposphere/new/2009/181/roap1810.09zpd.gz

The ZTD in this file is given in mm of delay. Thus, it is converted to m to compare with gLAB results

```
grep ROAP roap1810.09zpd|gawk -F\: '{print $3}'| gawk '{print $1,$2/1000}' > roap_igs.trp
```

Example 3: Zenith Troposphere Delay estimation

Tropospheric delay

The troposphere is the atmospheric layer situated between the Earth's surface and an altitude of about 60 km.

The effect of the troposphere on GNSS signals appears as an extra delay in the measurement of the signal travelling from satellite to receiver.

The tropospheric delay does not depend on frequency and affects both the pseudo-range (code) and carrier phases in the same way. It can be modeled by:

- A hydrostatic component, composed of dry gases (mainly nitrogen and oxygen) in hydrostatic equilibrium. This component can be treated as an ideal gas. Its effects vary with the temperature and atmospheric pressure in a reasonably predictable manner, and it is responsible for about 90% of the delay.
- A wet component caused by the water vapor condensed in the form of clouds. It depends on the weather conditions and varies faster than the hydrostatic component and in a totally random way. For high accuracy positioning, this component must be estimated together with the coordinates and other parameters in the navigation filter.

Backup

Overview

1. Introduction
2. The gLAB tool suite
3. Examples of GNSS processing using gLAB

➢ **Laboratory session organization**

LABORATORY Session

- Starting up your laptop
- Basic: Introductory lab exercises: Iono & Posit, SF, storm,TIDs
- Medium: Laboratory Work Projects: LWP1, LWP2
- Advanced: Homework

Laboratory session organization

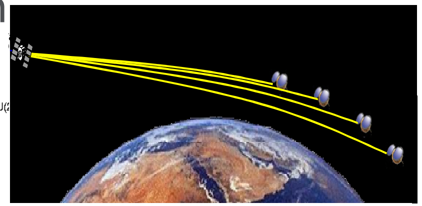
- The laboratory session is organized as an **assisted activity** where a set of **exercises must be developed individually or in groups of two**.
- As they are conceived as self-learning work, a detailed guide is provided in the slides (pdf file) to carry out the exercises.
- A notepad file with the command line instructions is also provided to help the sentence writing (doing copy & paste).
- A set of questions is presented, and the answers are also included in the slides.
- Teachers will attend individual (or collective) questions that could arise during exercise resolution.

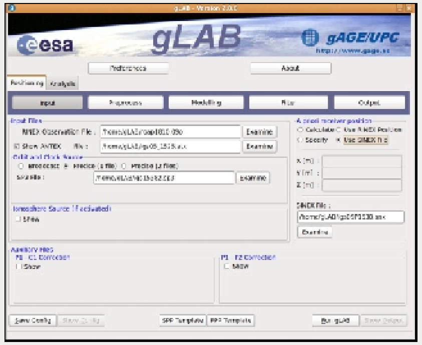
Laboratory session organization

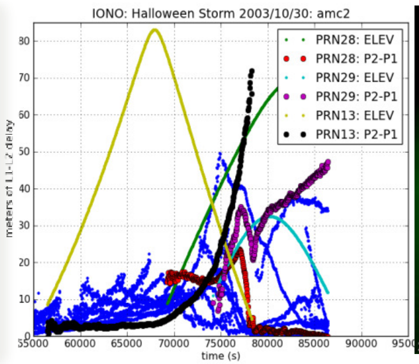
The exercises are organized at three different levels of difficulty. The student can choose the level of exercises to do, although at least one introductory exercise is recommended to learn basic gLAB usage.

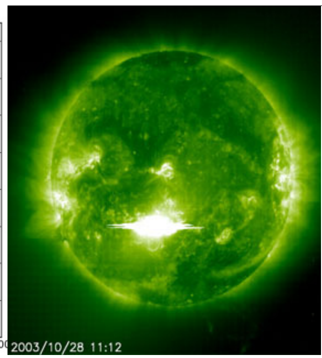
1. Basic: Introductory exercises.

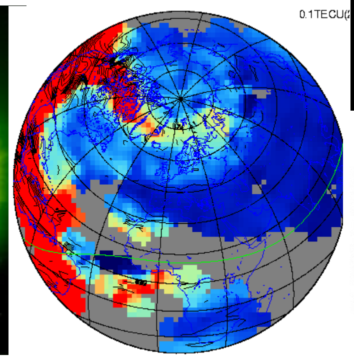
They consist of simple exercises to: 1) Study the Ionosphere effects on single frequency positioning. 2) To depict the STEC on a Radio occultation 3) Solar Flare effect on TEC, 4,5) TEC evolution during the Halloween Storm, 6) To depict a TID propagaction

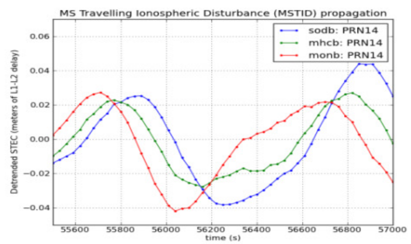










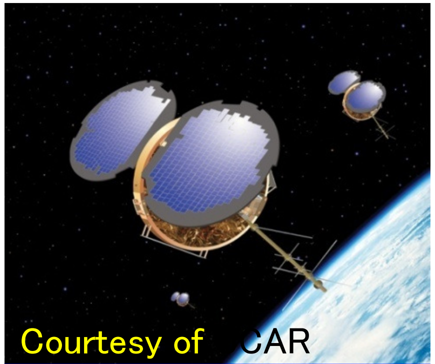


Laboratory session organization

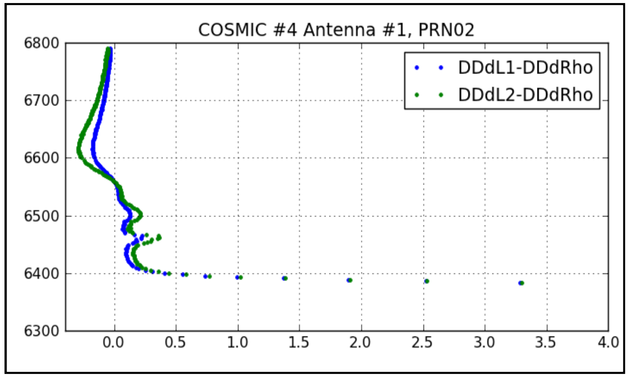
2. <u>Medium</u>: Laboratory Work Projects (LWP).

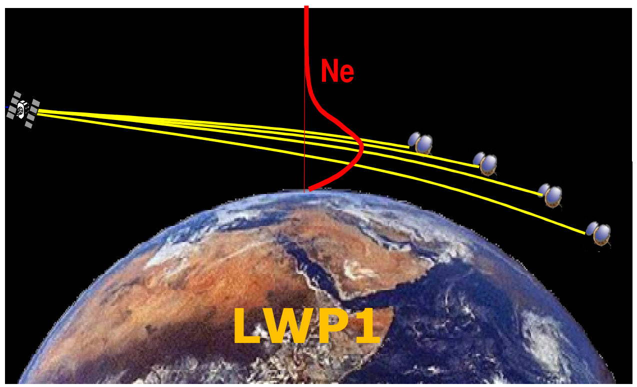
Two different LWP are proposed (<u>to choose from</u>):

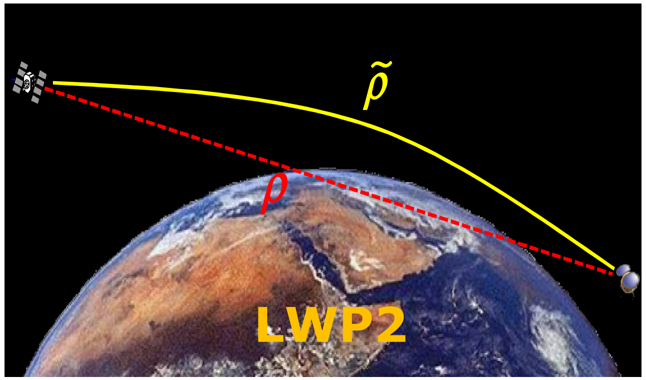
- **LWP1**: To show a simple numerical method to estimate electron density profiles (Ne(h)) from RO data.
- **LWP2**: To analyse the phase excess rate from GPS to LEO due to atmospheric (iono .& tropo.) bending.



Actual measurements from FORMOSAT-3/COSMIC LEOs will be used







A minimum knowledge of UNIX (e.g., awk) would be desirable

Laboratory session organization

3 . **Advanced:** Labeled as "Homework exercise".

The study ofcode-carrier ionosphere divergence effect on single-frequency carrier smoothed code is proposed as a complementary "Homework Exercise"

These exercises are beyond the scope of this 2h laboratory session, but can be selected, as well, instead of the Laboratory Work Projects (LWP1 or LWP2).

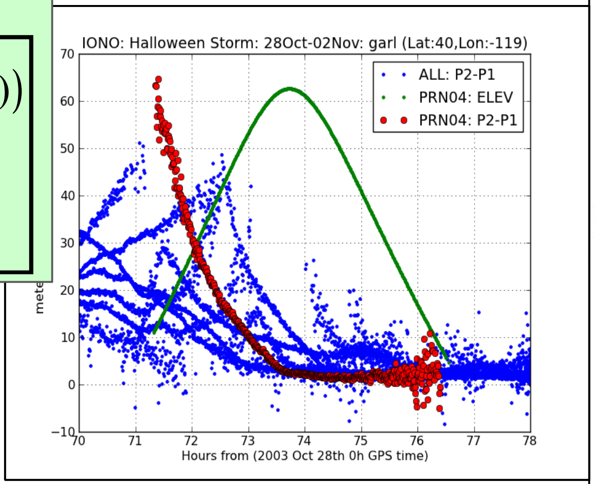
Hatch filter

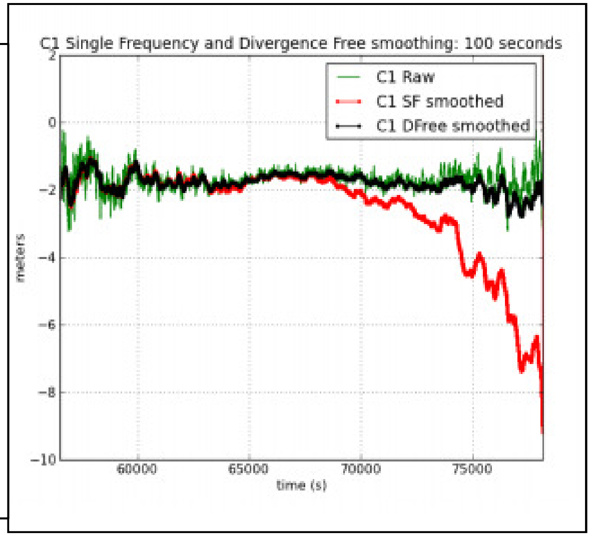
$$\hat{P}(k) = \frac{1}{n} P(k) + \frac{n-1}{n}\left(P(k-1) + L(k) - L(k-1)\right)$$

where $[n = k \text{ if } k < N]$, and $[n = N \text{ if } k \geq N]$.

The algorithm is initialised with: $\hat{P}(1) = P(1)$.

A minimum knowledge of UNIX (e.g., awk) is desirable for these homework exercises.






```
gawk 'BEGIN{g=(77/60)^2}{print $6, $4, (g*($13-$14)-($15-$16))/(g-1)}' meas.txt > PC.txt
```

Overview

1. Introduction
2. The gLAB tool suite
3. Examples of GNSS processing using gLAB
4. Laboratory session organization

LABORATORY Session

- **Starting up your laptop**
- Basic: Introductory lab exercises: Iono & Posit, SF, storm,TIDs
- Medium: Laboratory Work Projects: LWP1, LWP2
- Advanced: Homework

Starting up your laptop

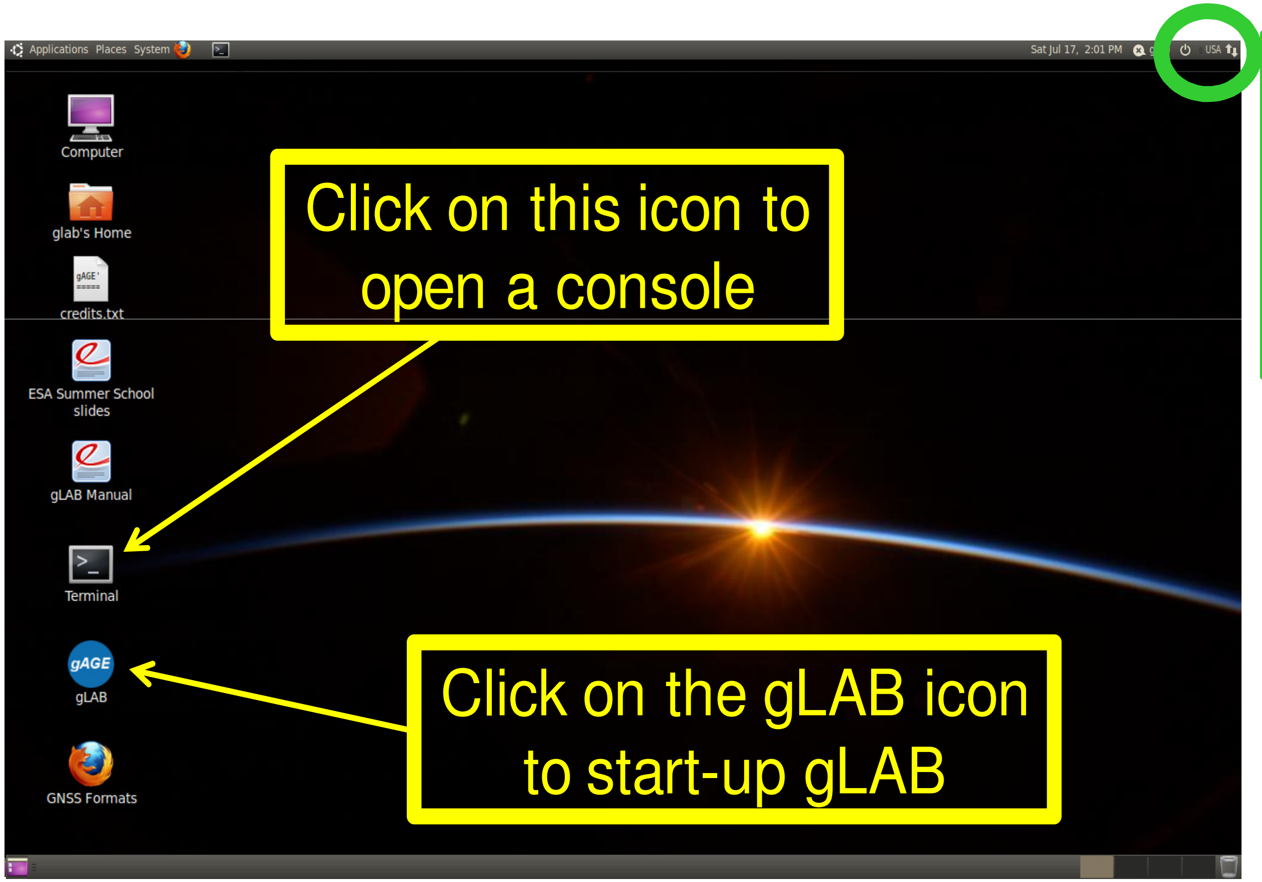
1. Plug the stick into an USB port and *boot your laptop from the stick*.

2. Access the Boot Device Menu when starting up the laptop.

Note: The way to do it *depends on your computer*: Usually, you should press **[ESC]** or **[F4]**, **[F10]**, **[F12]**....

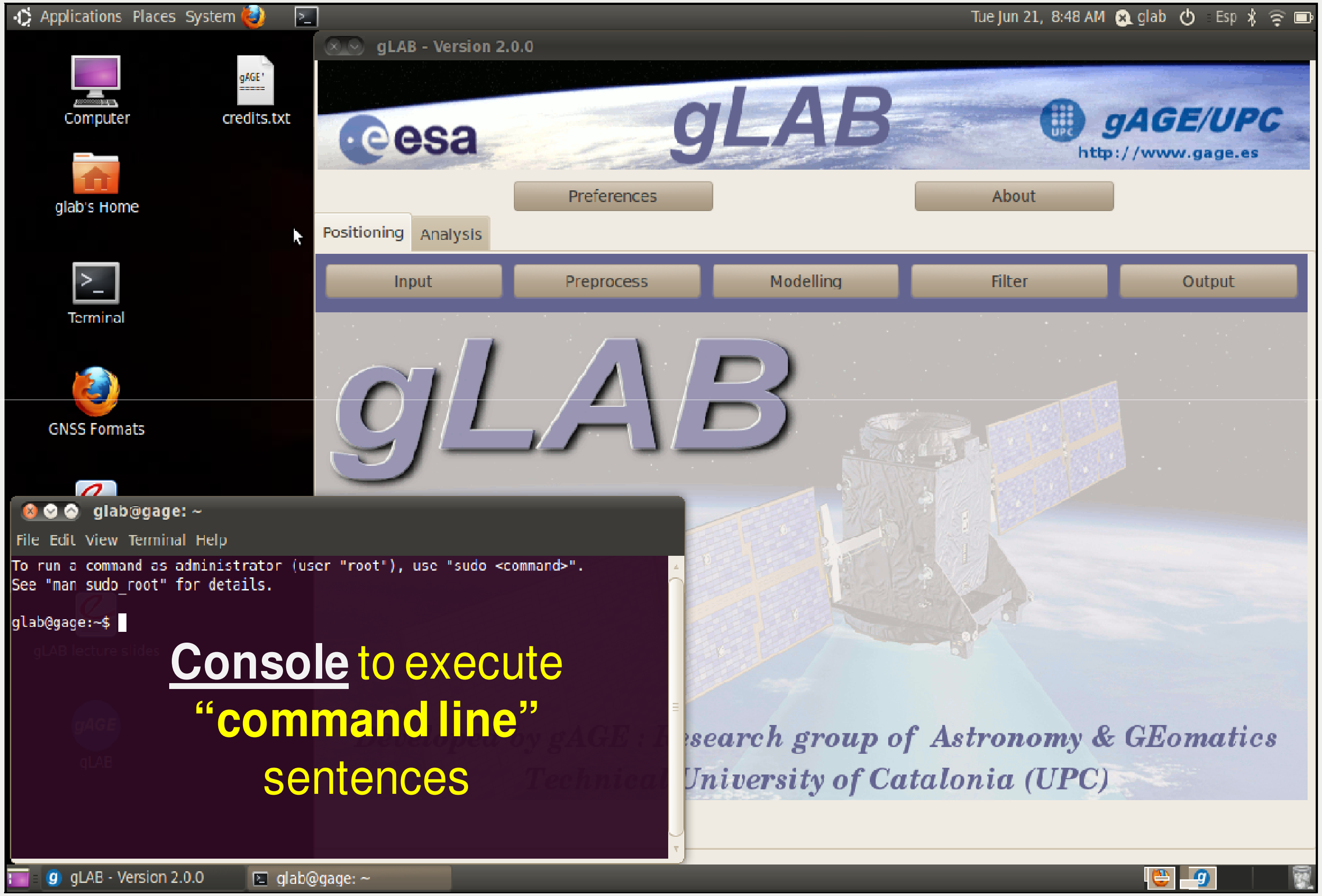
Starting up your laptop

3. The following screen will appear after about 2 minutes:



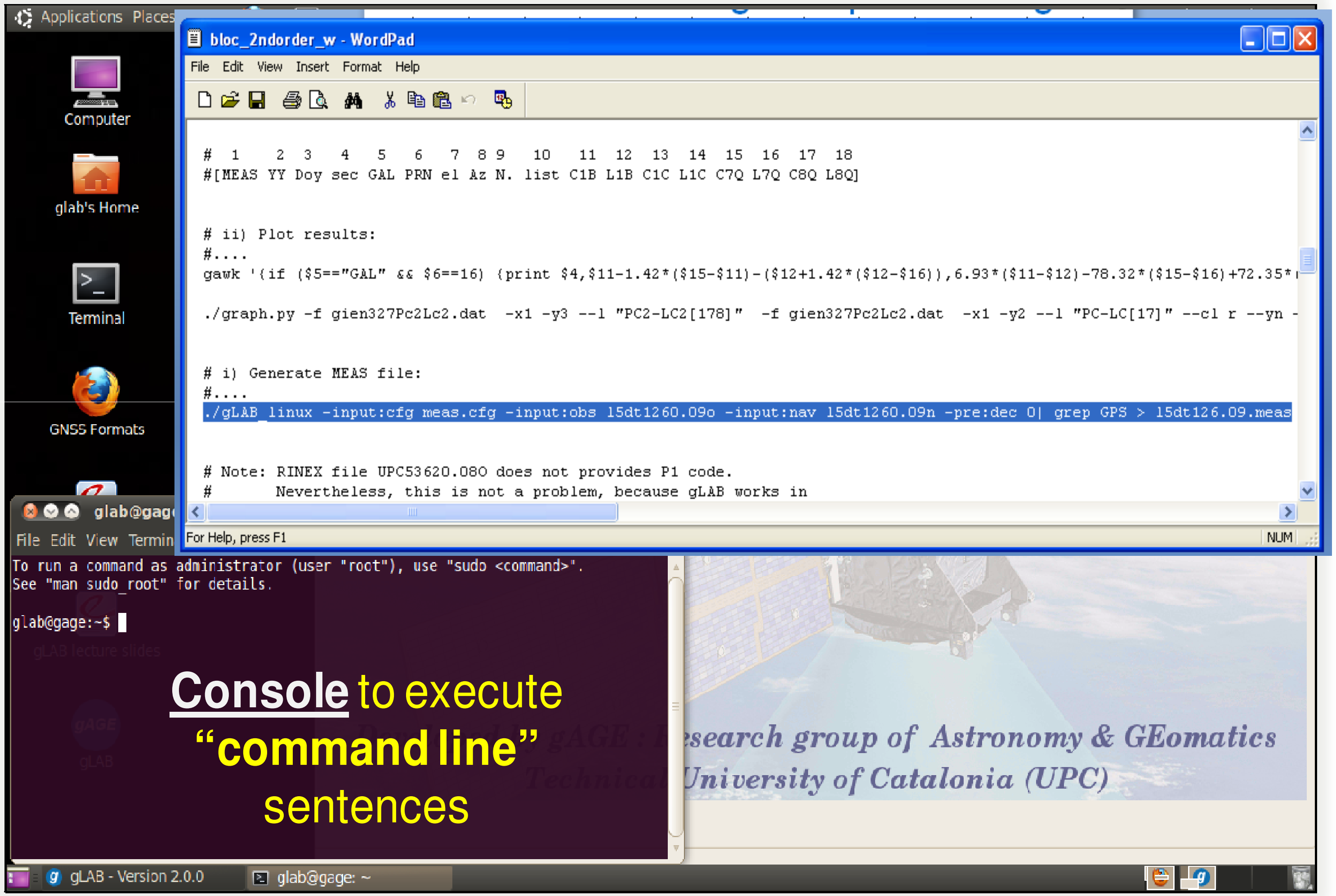
The US keyboard is set by default.
You can change it by clicking on the upper right corner.

Starting up your laptop



Now, the system is ready to start working!

Starting up your laptop



Copy and paste the sentences from notepad to console

Overview

1. Introduction
2. The gLAB tool suite
3. Examples of GNSS processing using gLAB
4. Laboratory session organization

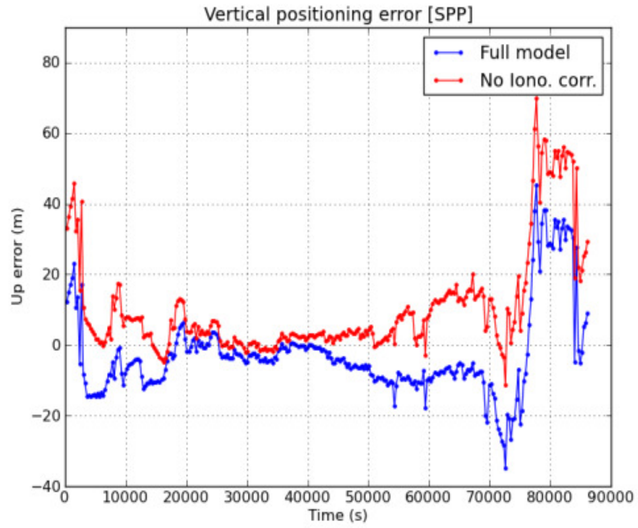
LABORATORY Session

- Starting up your laptop
- **Basic: Introductory lab exercises:** Iono & Posit, SF, storm,TIDs
- Medium: Laboratory Work Projects: LWP1, LWP2
- Advanced: Homework

EX. 1 Halloween storm: October 2003

A severe ionospheric storm was experienced on October 29-31, 2003 producing and increase of the electron density which led to large ionospheric refraction values on the GPS signals. Such conditions were beyond the capability of the GPS Klobuchar model broadcast for single frequency users, producing large errors in the SPS (see details in [R-3]).

Dual frequency users, navigating with the ionospheric-free combination of GPS signals were not affected by such ionospheric errors, as the ionospheric refraction can be removed up to 99:9% using dual-frequency signals.



Ex. 1: Assessing Iono effects on single freq. pos.

Exercise: Repeat the previous study of Example 1 to analyze the single frequency solution, but for the Halloween storm.

The following steps are recommended:

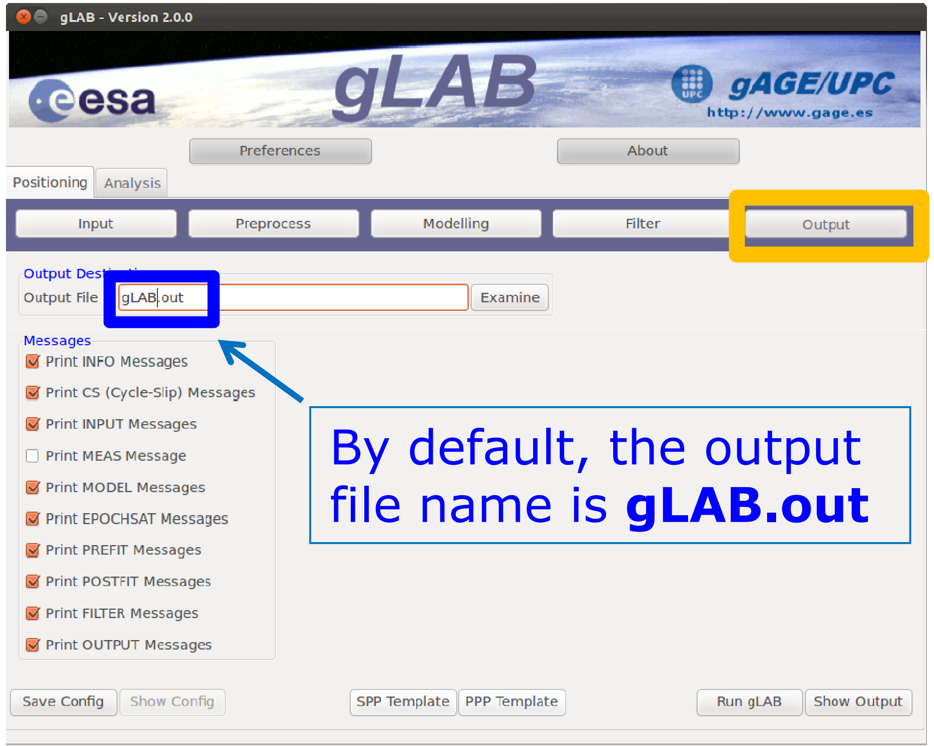
1. Using files **amc23020.03o,brdc3030.03n** compute with gLAB the following solutions:
   a) Solution with full SPS modeling. Name output file as: **gLAB.out**
   b) Solution with the ionospheric corrections disabled ➔ **gLAB1.out**
   c) Solution with the 2-freq. Ionosphere-free code (PC) ➔ **gLAB2.out**
2. Plot results

Note: The gLAB GUI or the command line sentences can also be used.
A "notepad" with the command line sentence is provided to facilitate the sentence writing: just "copy" and "paste" from notepad to the working terminal.

Tutorial associated to the **GNSS Data Processing book**
J. Sanz Subirana, J.M. Juan Zornoza, M. Hernández-Pajares

Ex. 1a: Full processing → gLAB.out

1. Compute SPP using files: `amc23020.03o,brdc3030.03n`



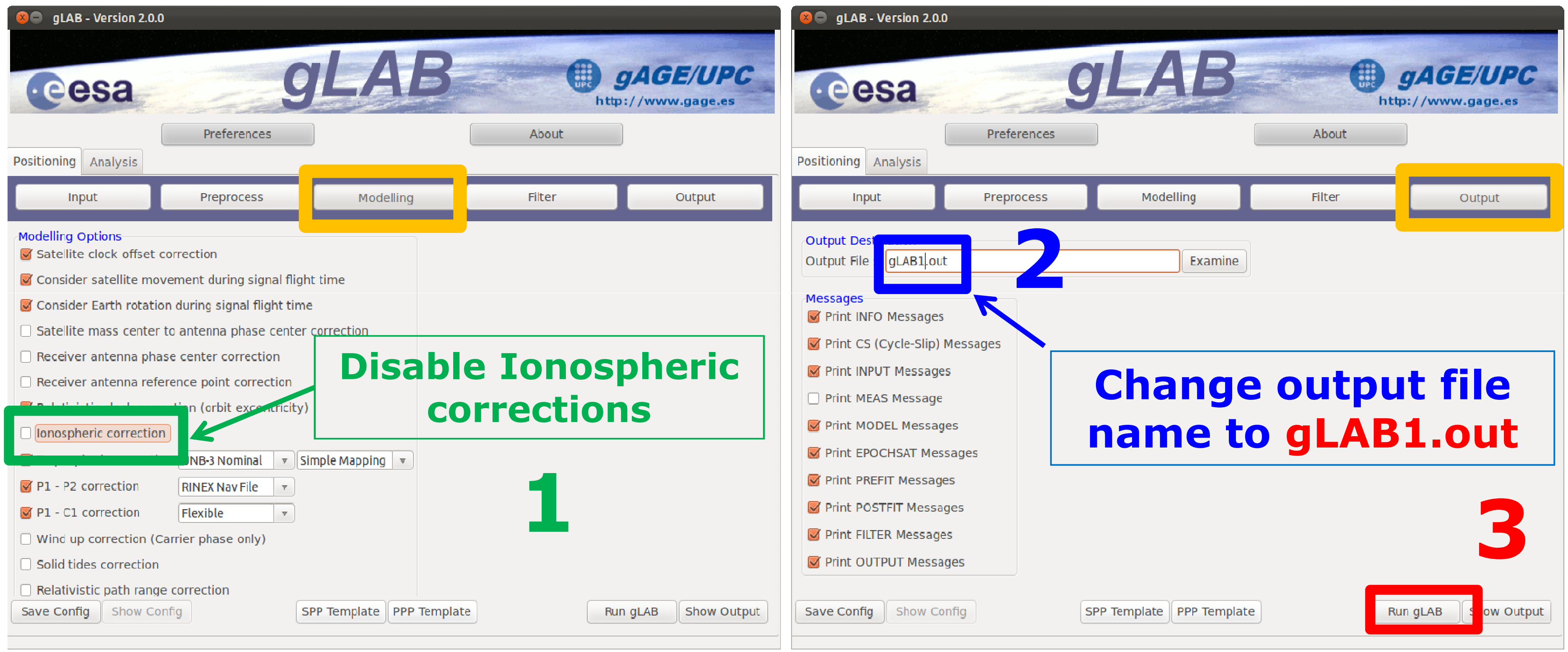
By default, the output file name is **gLAB.out**

Equivalent command line sentence:

```
gLAB_linux -input:cfg gLAB_p1_Full.cfg
           -input:obs amc3030.03o
           -input:nav brdc3030.03n
```

Ex. 1b: Iono disabled→ gLAB1.out

2. Reprocess the same files, with the **iono. corrections disabled**

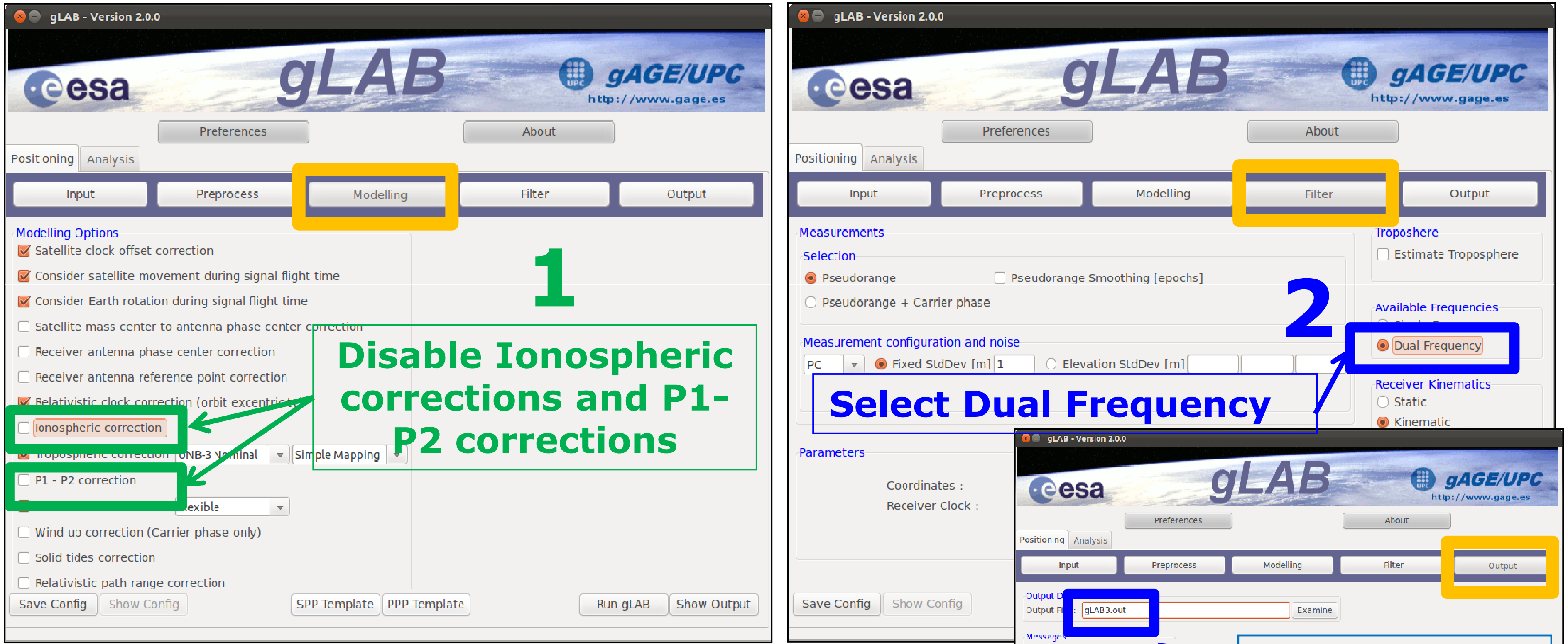


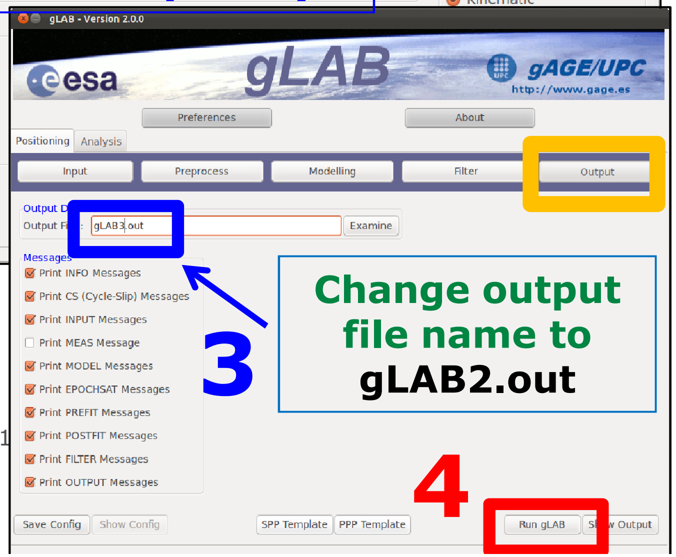
Equivalent command line sentence:

```
gLAB_linux -input:cfg gLAB_p1_NoIono.cfg
           -input:obs amc3030.03o
           -input:nav brdc3030.03n
```

Ex. 1c: 2 Freq processing → gLAB2.out

3. Reprocess the same files, but **with 2-freq. Iono.-free (PC)**





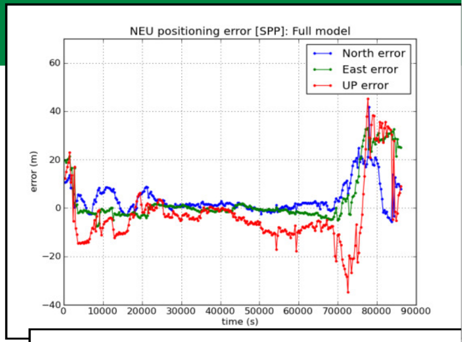
Equivalent command line sentence:

```
gLAB_linux -input:cfg gLAB_p1_IFree.cfg
           -input:obs amc3030.03o
           -input:nav brdc3030.03n
```

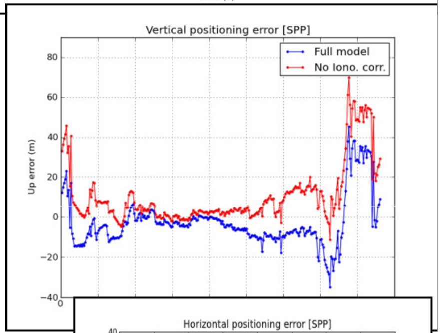
Ex. 1: Assessing Iono. effects on single freq. pos.

Execute in a single line: (or use the gLAB GUI)

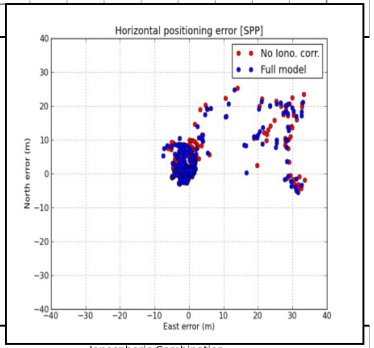
```
graph.py -f gLAB.out -x4 -y18 -s.- -c '($1=="OUTPUT")' -l "North error"
         -f gLAB.out -x4 -y19 -s.- -c '($1=="OUTPUT")' -l "East error"
         -f gLAB.out -x4 -y20 -s.- -c '($1=="OUTPUT")' -l "UP error"
                     --yn -40 --yx 70 --xl "time (s)" --yl "error (m)"
                     -t "NEU positioning error [SPP]: Full model"
```




```
graph.py -f gLAB.out  -x4 -y20 -s.- -c '($1=="OUTPUT")' -l "Full model"
         -f gLAB1.out -x4 -y20 -s.- -c '($1=="OUTPUT")' -l "No Iono." --cl r
                   --yn -40 --yx 90 --xl "Time (s)" --yl "Up error (m)"
                   -t "Vertical positioning error [SPP]"
```

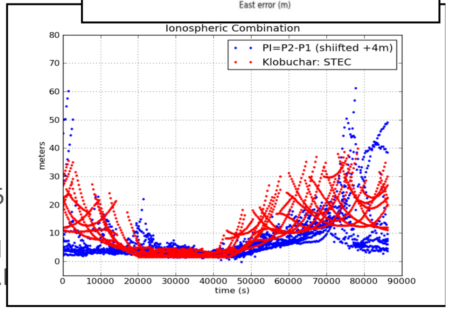



```
graph.py  -f gLAB1.out -x19 -y18 -so -c '($1=="OUTPUT")' -l "No Iono." --cl r
          -f gLAB.out  -x19 -y18 -so -c '($1=="OUTPUT")' -l "Full mod" --cl b
                   --xl "East error (m)" --yl "North error (m)"
                   --xn -40 --xx 40 --yn -40 --yx 40
                   -t "Horizontal pos. error [SPP]"
```



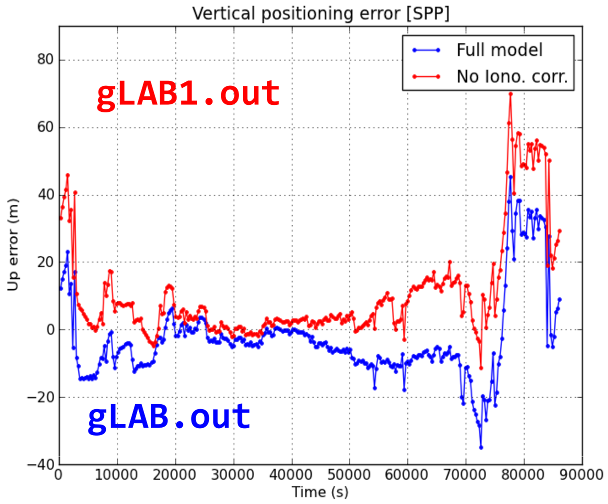
P2-P1 shifted +4 m

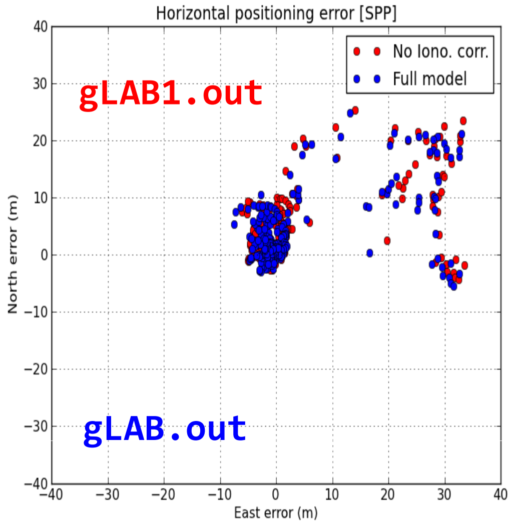
```
graph.py -f gLAB.out -x4 -y'($10-$9+4)' -s. -c '($1=="INPUT")'
         -f gLAB.out -x4 -y23 -s. -c '($1=="MODEL")' --cl r
                     --xl "time (s)" --yl "meters"
                     --yn 0 --yx 80  -t "Ionospheric Combination"
```

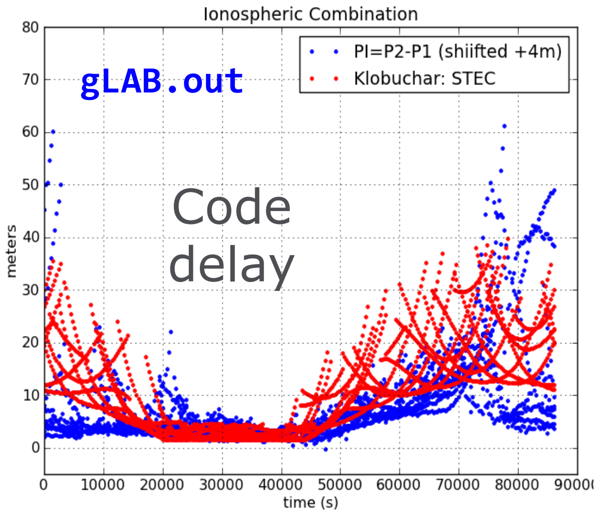


d
J. Sanz Subirana, J.

Ex. 1: Assessing Iono. effects on single freq. pos.







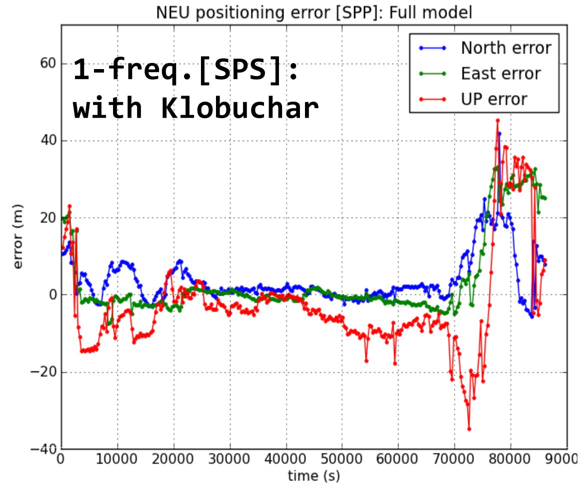
Ionospheric correction

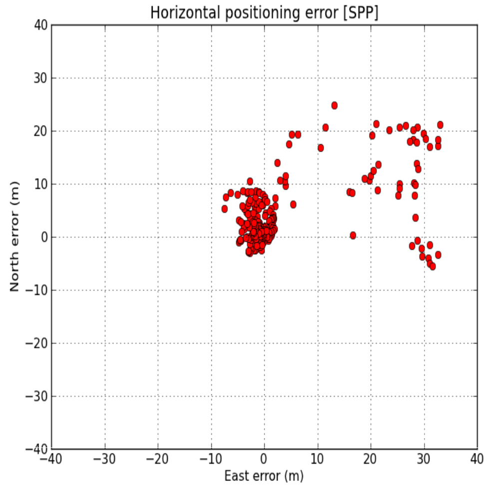
(broadcast Klobuchar)

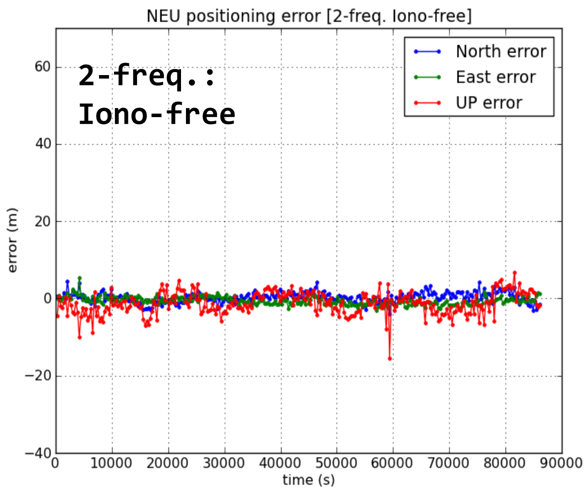
Ionospheric delays are larger at noon due to the higher insulation.

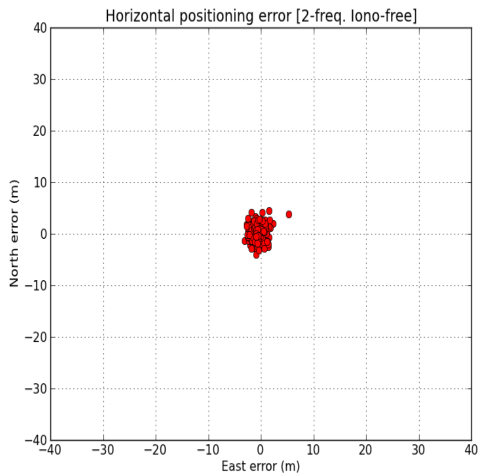
Klobuchar model is unable to mitigate the large ionospheric errors during the storm. Position domain errors reach up to 40 meters with Klobuchar corrections used.

Ex. 1: Assessing Iono. effects on single freq. pos.









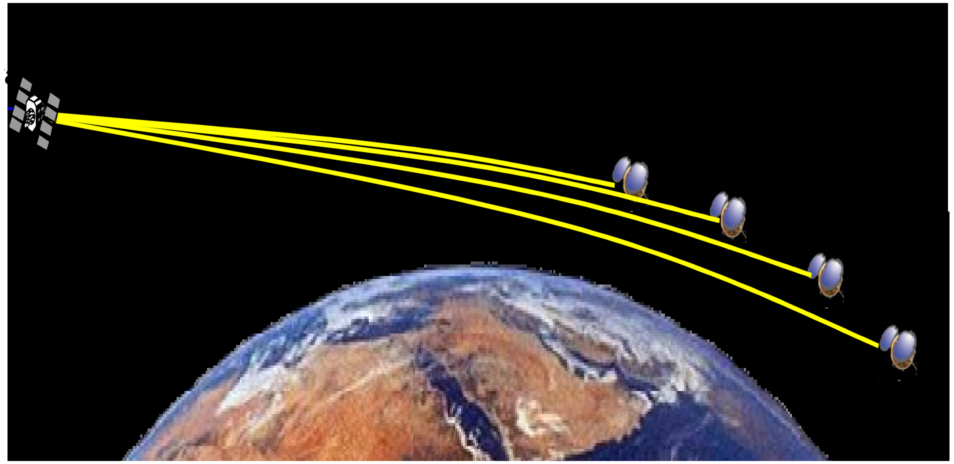
Ionospheric correction
(broadcast Klobuchar)

- The ionosphere-free combination (PC) of P1 and P2 codes is immune to the ionospheric storm.
- Although PC is three-times noisier than P1 or P2, it provides positioning accurate at the level of a few meters during the storm.

$$P_C = \frac{f_1^2 P_1 - f_2^2 P_2}{f_1^2 - f_2^2}$$

Ex. 2: STEC in a Radio Occultation (RO)

Radio Occultation (RO) commonly refers to a sounding technique in which a radio signal from a transmitting satellite (e.g., GPS) passes through a planetary atmosphere before arriving at a receiver on board a Low Earth Orbiter (LEO) satellite.



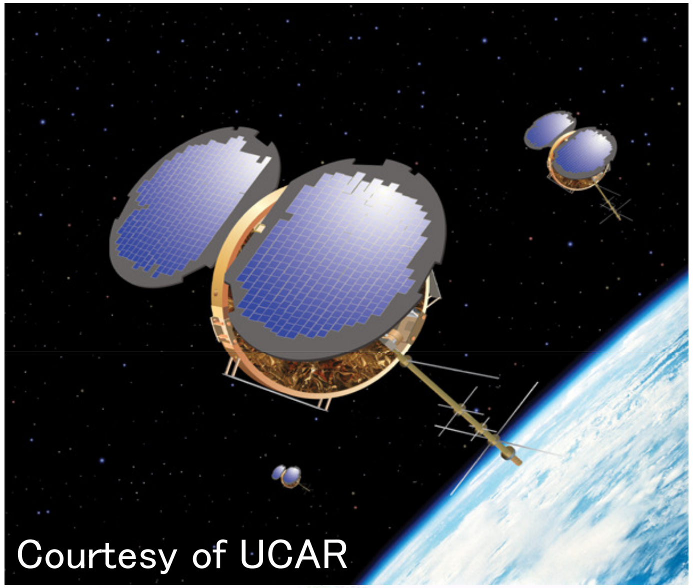
Along the ray path, the phase of the radio signal is perturbed in a manner related to the refractivity. RO measurements can reveal the refractivity, from which one can then derive atmospheric quantities such as Pressure, Temperature and the partial pressure of water vapor; and electron density, among others.

This is a simple exercise where the STEC variation along a radio occultation will be depicted using GPS L1, L2 measurements from a receiver on board a LEO of COSMIC constellation.

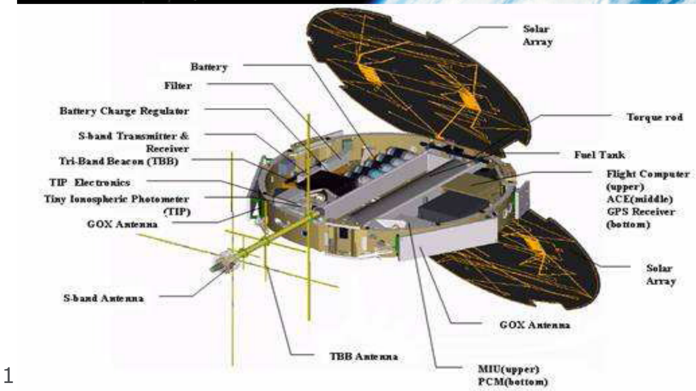
In the LWP1, Electron Density Profiles will be retrieved from this data using an algorithm equivalent to the Abel Transform.

FORMOSAT-3/COSMIC mission

Constellation Observing System for Meteorology Ionosphere and Climate:
6 microsatellites; orbit altitude ~ 800km



- **Three instruments:**
  GPS receiver. <u>4 antennas: 2 for POD, 2 for RO</u>.
  TIP, Tri-band beacon
- **Weather + Space Weather data.**
- **Global observations of:**
  Pressure, Temperature, Humidity
  Refractivity
  Ionospheric Electron Density
  Ionospheric Scintillation
- **Demonstrate quasi-operational GPS limb** sounding with global coverage in near-real time
- **Climate Monitoring**



Information available at www.cosmic.ucar.edu

Ex. 2: STEC in a Radio Occultation (RO)

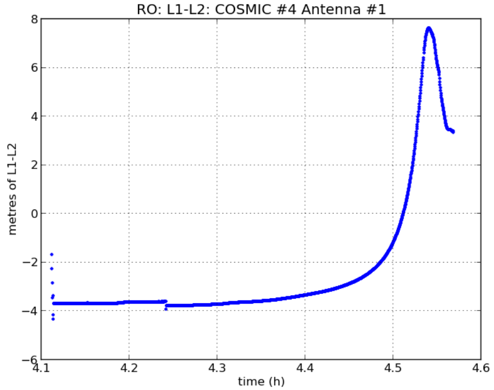
Exercise: The file "RO.obs" contains the following fields [*]:

```
                             |←------ LEO ------>|<------- GPS ------->|
YY DoY HH.HH CODE PRN  elev  r_LEO AR_LEO DEC_LEO  r_GPS AR_GPS DEC_GPS   L1  L2    L1-L2 arc
                       (deg)  (km) (Deg)   (Deg)    (km)  (Deg)  (Deg)   (cycles)    (m)
 1   2    3    4    5    6      7    8       9       10     11     12      13  14     15   16
```

Plot the L1-L2 measurement in function of time to depict the variation of STEC along the occultation:

➔ Select for instance: **PRN=02** and "**CODE=1241**", that corresponds to **LEO=4** and Antenna **1**

```
- Selecting: CODE=1241 and PRN=02
  grep 1241 RO.obs|gawk '{if ($5==02) print $3,$15}'> ro.dat

- Ploting L1-L2
  graph.py -f  ro.dat --xl "time (h)"
   --yl "meters of L1-L2"  -t"RO: L1-L2: COSMIC #4 Antenna #1"
```



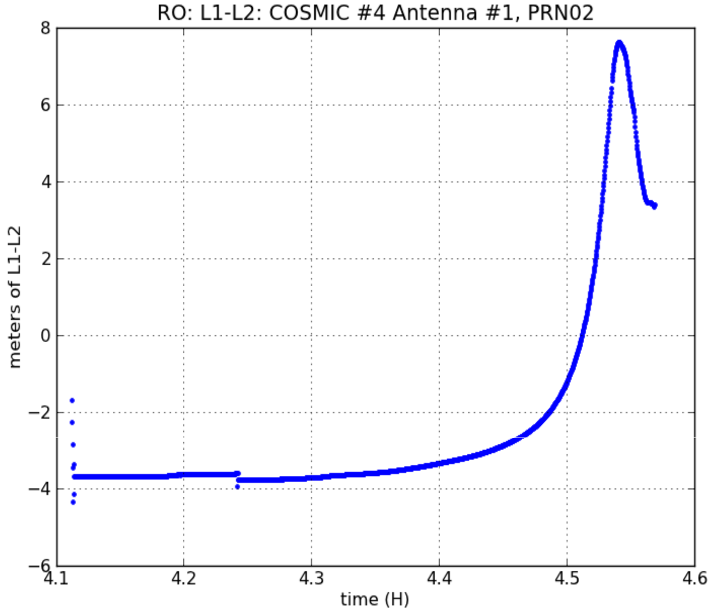
[*] This file has been generated from files (http://cosmic-io.cosmic.ucar.edu/cdaac):
podObs_2006.253.004.01.01_rnx, leoOrb_2006.253.004.01_2009.2650_sp3, igs13920.sp3

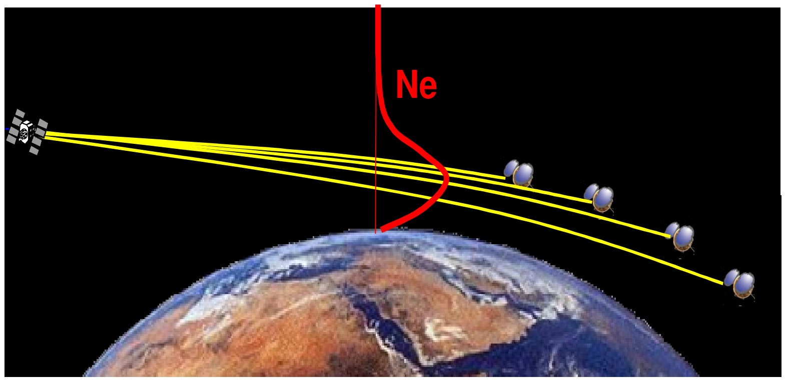
Ex. 2: STEC in a Radio Occultation (RO)

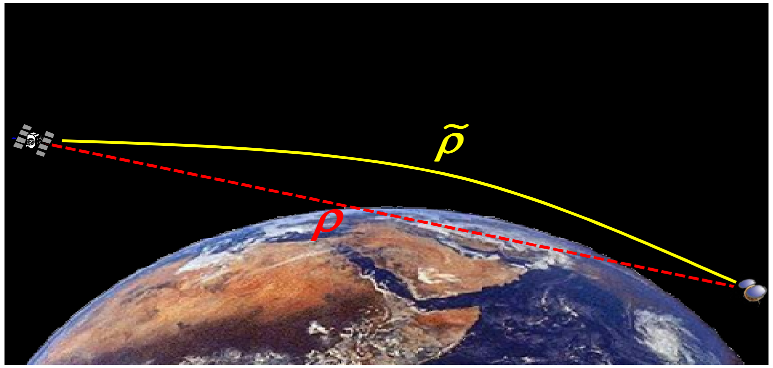
The previous plot shows only the variation of the Integrated Electron Content along the ray path (STEC).

More information can be retrieved from occultation measurements. For instance:

- Electron Density Profile of the Ionosphere (LWP1).
- Phase excess rate, which is related to the bending of ray (LWP2).





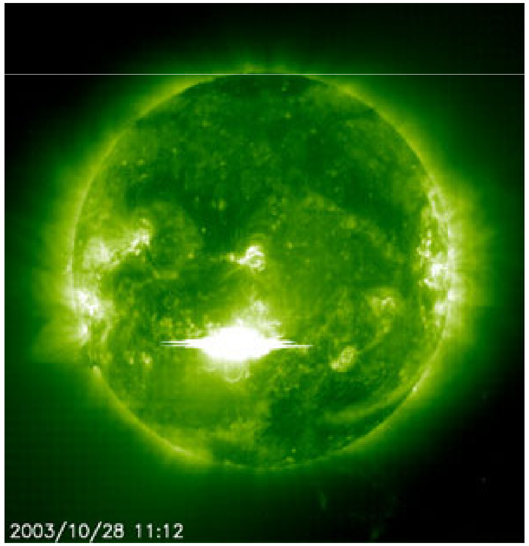


Given that session time is limited to 2h, participants who feel comfortable using gLAB, can skip part of the next basic exercises (Ex3..., Ex6) and jump to the Laboratory Work Projects (LWP).

There, if you prefer, you can jump to slide #86 and choose one from the two LWPs

Ex. 3: Solar Flare October 28, 2003

On October 28, 2003, an intense solar eruption (a Solar Flare) was detected around 11h UT in an active region which had grown one of the largest sunspots ever seen by the SOlar Helioscopic Observatory (SOHO). It appeared as a bright spike in the SOHO ultraviolet images.



This sudden enhancement of the solar radiation in the X-ray and extreme ultra-violet band produced a sudden increase in the ionospheric electron density on the daylight hemisphere, (see [R-3]).

Ex. 3: Solar Flare October 28, 2003

Exercise: Analyze the effect of the Solar Flare on the Slant Total Electron Content (STEC) measurements of four permanent IGS receivers ankr, asc1, kour and qaq1, covering a wide range of longitude and latitude.

Data sets:

ankr3010.03o, asc13010.03o,

kour3010.03o, qaq13010.03o

Tutorial associated to the **GNSS Data Processing book**
J. Sanz Subirana, J.M. Juan Zornoza, M. Hernández-Pajares

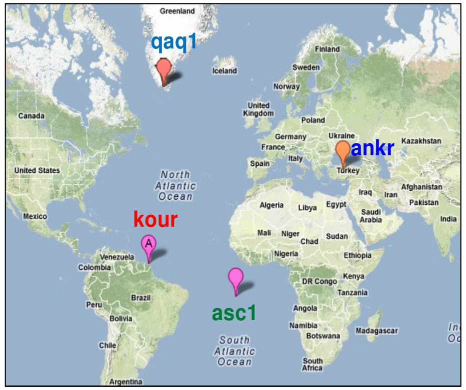
Ex. 3: Solar Flare October 28, 2003

[Id	YY	Doy	sec	GPS	PRN	el	Az	N.	list	C1C	L1C	C1P	L1P	C2P	L2P]
1	2	3	4	5	6	7	8	9	10	11	xx	13	14	15	16]

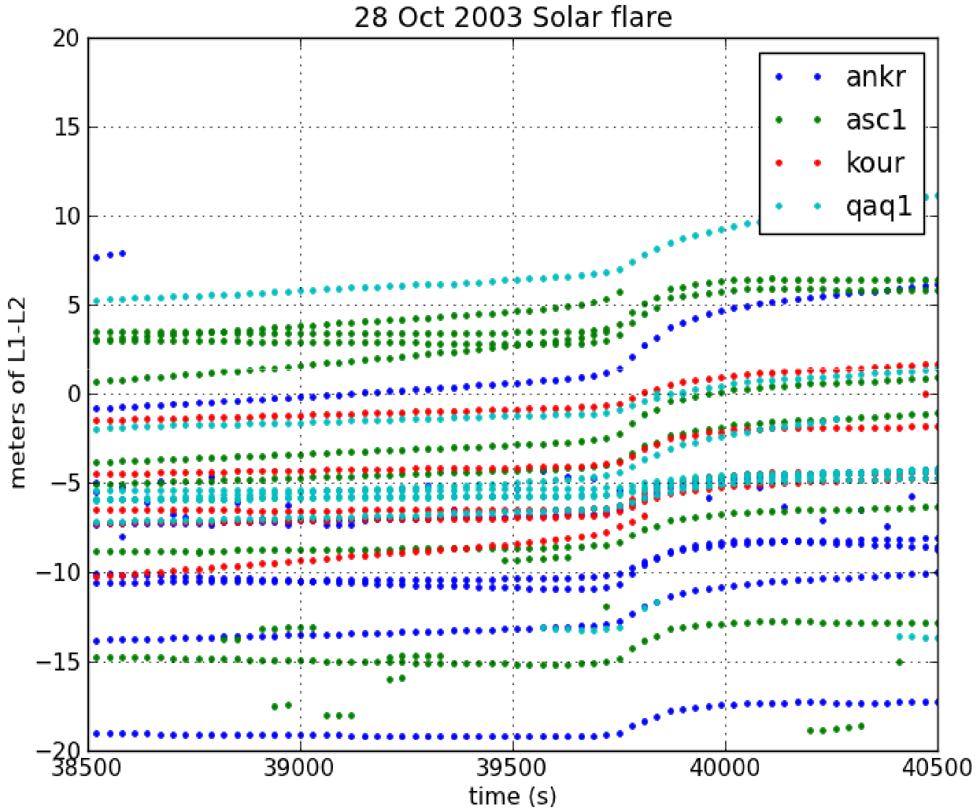
Execute:

```
gLAB_linux -input:cfg meas.cfg -input:obs ankr3010.03o > ankr3010.03.meas
gLAB_linux -input:cfg meas.cfg -input:obs asc13010.03o > asc13010.03.meas
gLAB_linux -input:cfg meas.cfg -input:obs kour3010.03o > kour3010.03.meas
gLAB_linux -input:cfg meas.cfg -input:obs qaq13010.03o > qaq13010.03.meas
```

```
graph.py -f ankr3010.03.meas -x4 -y'($14-$16)' -l "ankr"
         -f asc13010.03.meas -x4 -y'($14-$16)' -l "asc1"
         -f kour3010.03.meas -x4 -y'($14-$16)' -l "kour"
         -f qaq13010.03.meas -x4 -y'($14-$16)' -l "qaq1"
         --xl "time (s)" --yl "meters of L1-L2"
         --xn 38500 --xx 40500 --yn -20 --yx 20
         -t "28 Oct 2003 Solar flare"
```



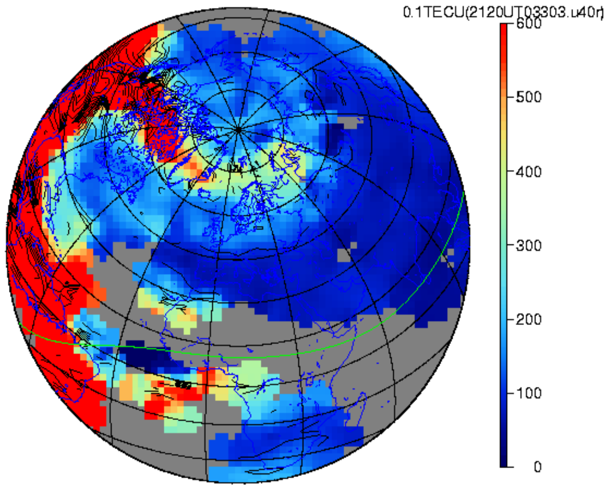
Ex. 3: Solar Flare October 28, 2003



Ex. 4: Halloween storm: P2-P1 analysis

Associated with the Solar Flare analysed in the previous exercise, a Coronal Mass Ejection occurred, which sent a large particle cloud impacting the Earth's magnetosphere about 19 hours later, on October 29.

Subsequent impacts were still occurring several hours later. This material interacted with the Earth's magnetosphere and a Storm Enhancement Density (SED) appeared in North America and affected later the northern latitudes in Europe. Extra large gradients of TEC associated with this phenomenon were also produced, degrading the GPS positioning performance.



The TEC evolution in October 30, 2003 (i.e., Day 303 of year 2003) can be seen in the movie **TEC_2003oct30_anim.gif**.

Ex. 4: Halloween storm: P2-P1 analysis

The measurement files `garl3010.03o`, `garl3020.03o`, `garl3030.03o`, `garl3040.03o`, `garl3050.03o`, `garl3060.03o` were collected by the permanent receiver **`garl`** in Empire, Nevada, USA (ϕ= 40.42 deg, λ =-119.36 deg) from October 28 to November 2, 2003.

Using these files, plot the STEC for all satellites in view and discuss the range of such variations. Analyse, in particular, the satellite PRN 04 and calculate the maximum rate of STEC variation in mm/s of L1 delay. Add the elevation of satellite PRN 04 in the plot.

The associated broadcast navigation les are brdc3010.03n, brdc3020.03n, brdc3030.03n, brdc3040.03n, brdc3050.03n, brdc3060.03n.

Ex. 4: Halloween storm: P2-P1 analysis

Exercise: Depict the ionospheric delays for the different satellites in view from station amc2.

- This is a simple exercise aimed to illustrate how to use gLAB to easily analyze GNSS measurements and their combinations.
- gLAB will be used to read the RINEX measurements file and to generate a "text" with the measurements provided in a columnar format (more suitable to make plots).
- Using such "text" file, the STEC pattern for the different satellites in view during the storm is depicted from the geometry-free combination of codes *P2-P1.*

Note: $P_2 - P_1 = I + K_{21}$

Ex. 4: Halloween storm: P2-P1 analysis

The next commands read a RINEX file and generate a text file (in columnar format) that allows to easily plot the **measurements and their combinations**:

1. Using the configuration file **meas.cfg**, READ the RINEX and generate the MEAS message with data format:
   Execute:

[Id	YY	Doy	sec	GPS	PRN	el	Az	N.	list	C1C	L1C	C1P	L1P	C2P	L2P]	
1	2	3	4	5	6	7	8	9	10	11	xx	13	14	15	16	]

```
gLAB_linux -input:cfg meas.cfg -input:obs amc23030.03o -input:nav brdc3030.03n > amc23030.03.meas
```

2. From **meas.txt** file,

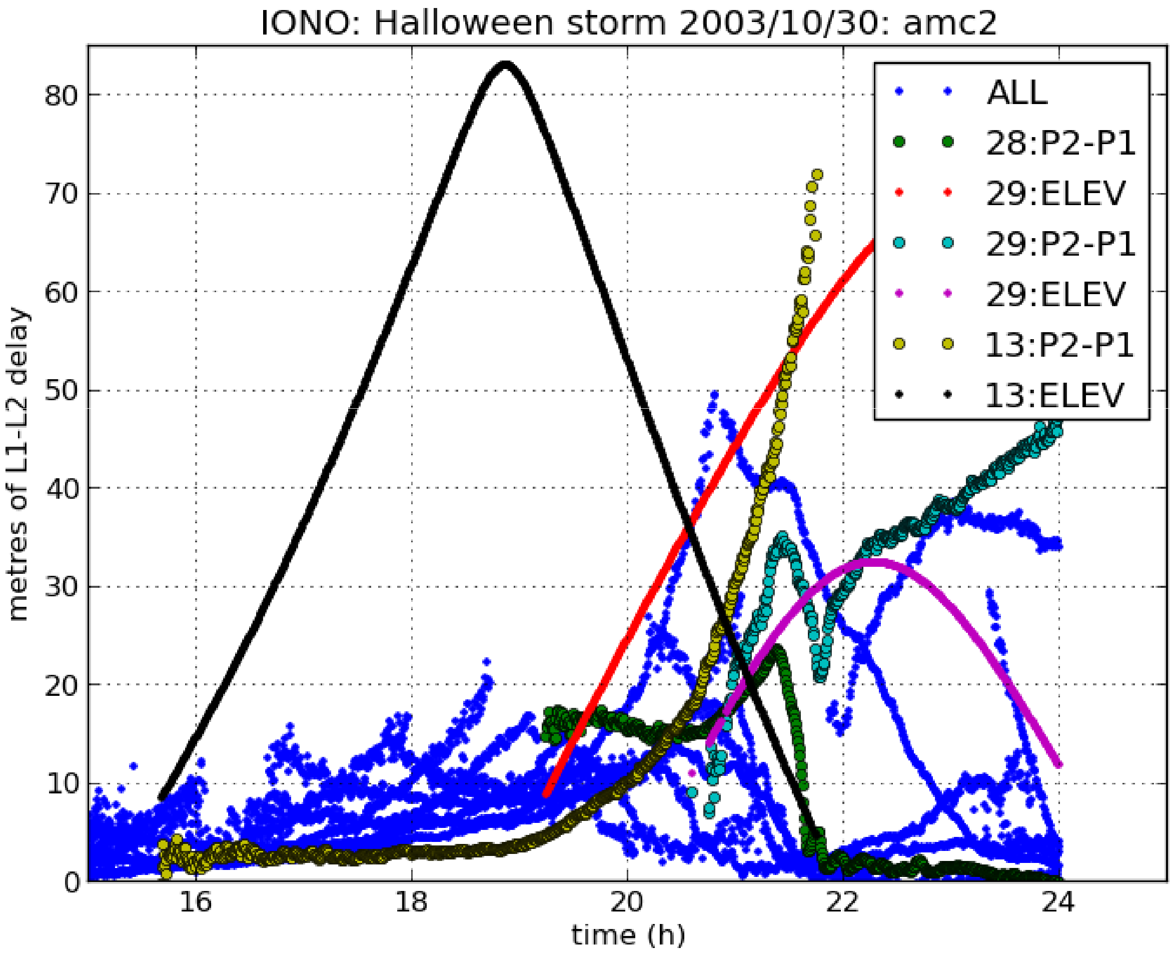
   Compute the ionospheric combination of codes: *PI=P2-P1.*
   Generate the file PI.txt with the following content: [PRN, hour, PI, elevation]

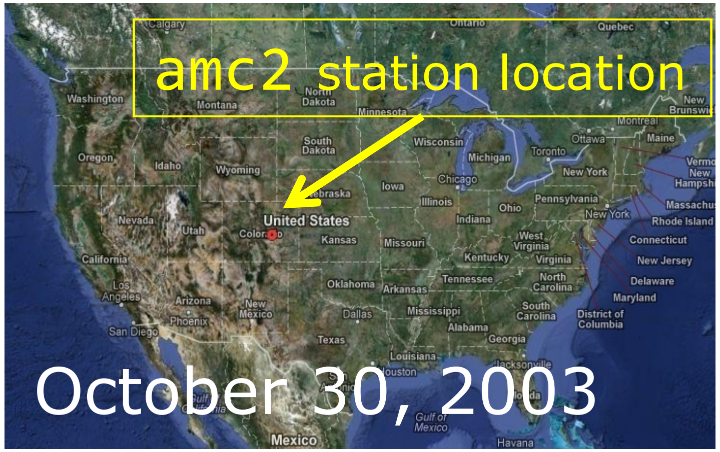
```
gawk '{print $6, $4/3600, $15-$13, $7}' amc23030.03.meas > PI.txt
```

3. From **PI.txt** file,
   Plot the *PI=P2-P1* for time interval [15 to 24].hours. Show in the same graph: 1) ALL satellites, 2) PRN 13, 28 and 29, and 3) The elevation of each satellite.(13, 28 and 29)

```
graph.py -f PI.txt -x2 -y3 -l "ALL"
   -f PI.txt -c'($1==28)' -x2 -y3 -so -l "28:P2-P1" -f PI.txt -c'($1==28)' -x2 -y4 -l "29:ELEV"
   -f PI.txt -c'($1==29)' -x2 -y3 -so -l "29:P2-P1" -f PI.txt -c'($1==29)' -x2 -y4 -l "13:ELEV"
   -f PI.txt -c'($1==13)' -x2 -y3 -so -l "13:P2-P1" -f PI.txt -c'($1==13)' -x2 -y4 -l "13:ELEV"
   --xn 15 --xx 25 --yn 0 --yx 85 --xl "time (h)" --yl "meters of L1-L2 delay"
```

Ex. 4: Halloween storm: P2-P1 analysis





Ex. 5: Halloween storm evolution

Exercise: Analyze the ionospheric delays for 6 consecutive days including the Halloween storm

- This is a simple exercise aimed to illustrate the ionospheric delays variation during the Halloween storm. A period of 6 consecutive days (from October 28 to November 2, 2003) are analyzed using measurements collected in the **"garl"** station in North America.

- The STEC variations are depicted from the geometry-free combination of codes *P2-P1.*

Note: $P_2 - P_1 = I + K_{21}$

Tutorial associated to the **GNSS Data Processing book**
J. Sanz Subirana, J.M. Juan Zornoza, M. Hernández-Pajares

Ex. 5: Halloween storm evolution

1.- Read RINEX file:

[Id	YY	Doy	sec	GPS	PRN	el	Az	N.	list	C1C	L1C	C1P	L1P	C2P	L2P]	
1	2	3	4	5	6	7	8	9	10	11	xx	13	14	15	16	]

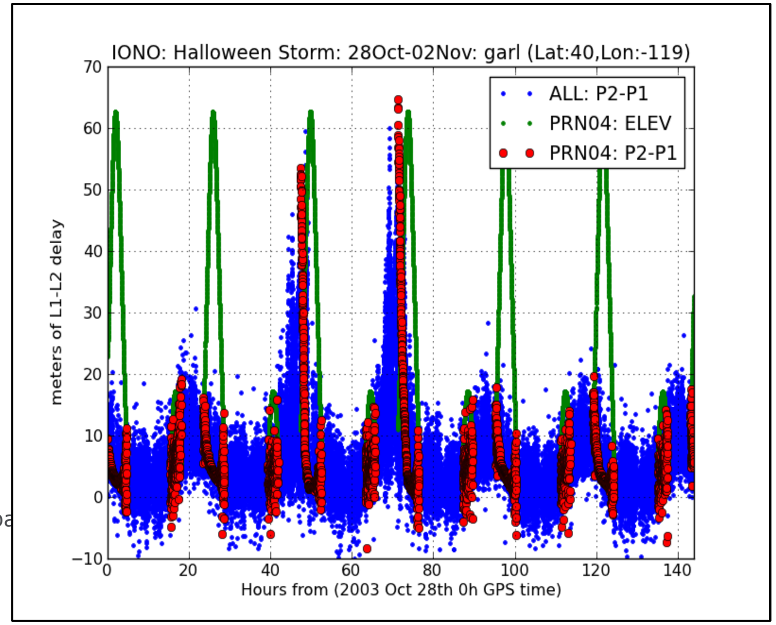
```
gLAB_linux -input:cfg meas.cfg -input:obs garl3010.03o -input:nav brdc3010.03n > garl3010.03.meas
gLAB_linux -input:cfg meas.cfg -input:obs garl3020.03o -input:nav brdc3020.03n > garl3020.03.meas
gLAB_linux -input:cfg meas.cfg -input:obs garl3030.03o -input:nav brdc3030.03n > garl3030.03.meas
gLAB_linux -input:cfg meas.cfg -input:obs garl3040.03o -input:nav brdc3040.03n > garl3040.03.meas
gLAB_linux -input:cfg meas.cfg -input:obs garl3050.03o -input:nav brdc3050.03n > garl3050.03.meas
gLAB_linux -input:cfg meas.cfg -input:obs garl3060.03o -input:nav brdc3060.03n > garl3060.03.meas
```

2.- Merge files and refer all the data to 0h of October 28th: Doy0301:

```
cat garl30?0.03.meas |gawk '{d=($3-301)*86400;$4=$4+d; print $6, $4/3600, $11-$13, $7}' >PI.txt
```

3.- Plot results:

```
graph.py -f PI.txt -x2 -y3 -l "ALL P2-P1"
         -f PI.txt -c'($1==04)' -x2 -y4 -s. -l "PRN04: ELEV"
         -f PI.txt -c'($1==04)' -x2 -y3 -so -l "PRN04: P2-P1"
         --xn 0 --xx 144
```

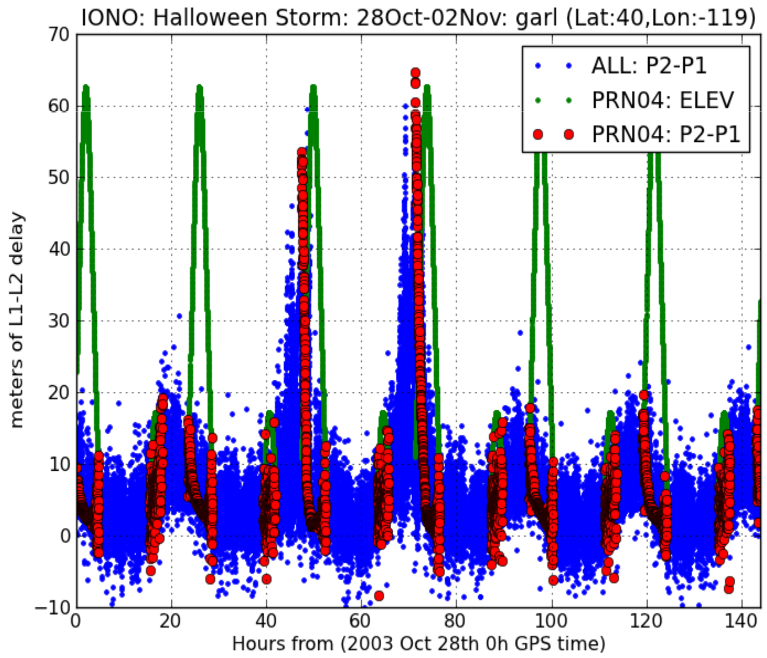


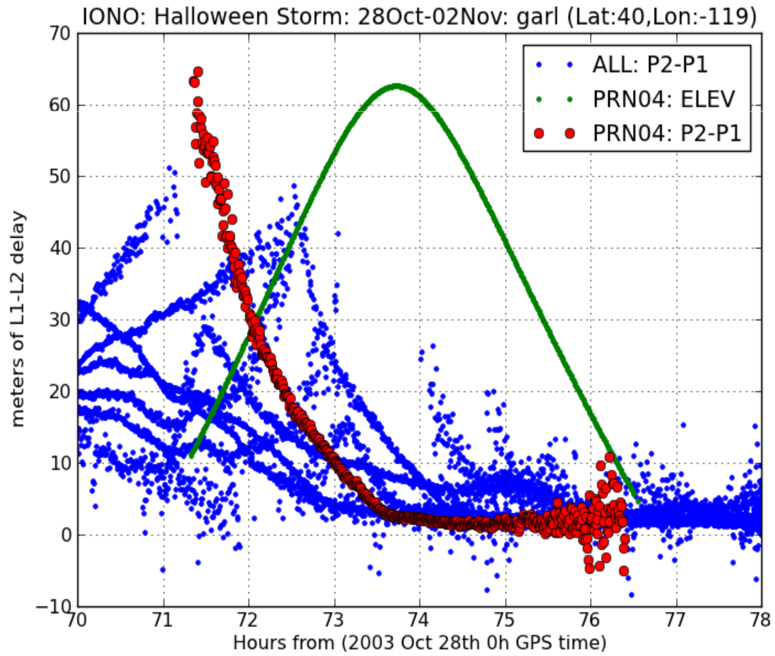
Ex. 5: Halloween storm evolution

Zoom at time interval: 70 to 78 h

```
graph.py -f PI.txt -x2 -y3 -l "ALL P2-P1"
  -f PI.txt -c'($1==04)' -x2 -y4 -l "04: EL"
  -f PI.txt -c'($1==04)' -x2 -y3 -so -l "04"
  --xn 0 --xx 144
```

```
graph.py -f PI.txt -x2 -y3 -l "ALL P2-P1"
  -f PI.txt -c'($1==04)' -x2 -y4 -l "04: EL"
  -f PI.txt -c'($1==04)' -x2 -y3 -so -l "04"
  --xn 70 --xx 78
```





Ex. 6: Travelling Ionospheric Disturb.

Travelling Ionospheric Disturbances (TIDs) are understood as plasma density fluctuations that propagate through the ionosphere at an open range of velocities and frequencies. The trend of such fluctuations can be seen from the geometry free combination of GPS carrier measurements $L_I = L_1 - L_2$.

Some authors distinguish between Large-Scale TIDs (LSTIDs) with a period greater than 1 hour and moving faster than *0,3 km/s*, and Medium- Scale TIDs (MSTIDs) with shorter periods (from 10 minutes to 1 hour) and moving slower (*0.05-0.3 km/s*). The LSTIDs seem to be related to geomagnetic disturbances (i.e., aurora, ionospheric storms, etc.). The origin of MSTIDs seems to be more related to meteorological phenomena such as neutral winds, eclipses, or solar terminator that produces Atmospheric Gravity Waves (AGW) being manifested as TIDs at ionospheric heights, due to the collision between neutral and ionised molecules.

Ex. 6: Travelling Ionospheric Disturb.

In [R4, 2006] a simple method to detect MSTIDs is proposed. It consists of detrending the geometry free combination of GPS carrier measurements from the diurnal variation and elevation angle dependences, applying the following equation:

$$\delta L_I(t) = L_I(t) - \left(L_I(t+\tau) - L_I(t-\tau) \right) / 2$$

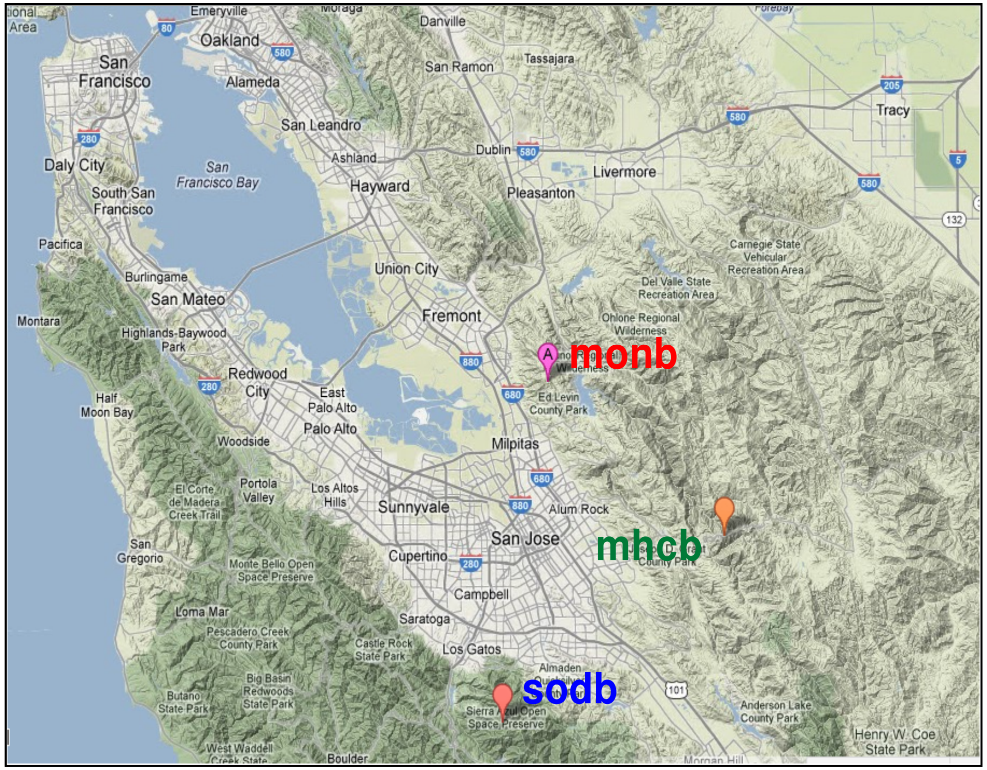
where a value of 300sec is suitable to keep enough variation of LI (i.e., STEC).

Using the previous equation, the detrending is done simply by subtracting from each value an average value of the previous and posterior measurements (i.e., the curvature of the LI temporal dependency). It must be pointed out that that this detrending procedure can be used in real time with a single receiver, so it is suitable for identifying these ionospheric perturbations in navigation applications.

Ex. 6: Travelling Ionospheric Disturb.

An example of MSTID propagation can be depicted as follows using the measurements of three stations SODB, MHCB and MONB, which are separated by a few tens of kilometres.

The target is to reproduce the figure 10 of the above mentioned paper [R4, 2006].



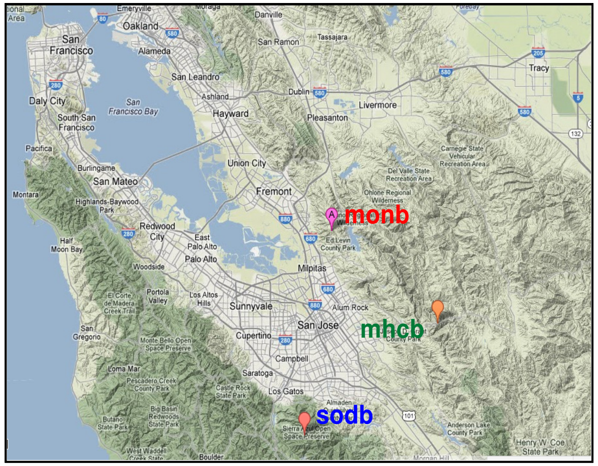
[R4, 2006] Hernández-Pajares, M; Juan, M.; Sanz, J., 2006].

Ex. 6: Travelling Ionospheric Disturb.

Exercise: Execute <u>in a single line</u>:

```
# a) Reading RINEX files:
gLAB_linux -input:cfg meas.cfg -input:obs mhcb2910.01o > mhcb.meas
gLAB_linux -input:cfg meas.cfg -input:obs monb2910.01o > monb.meas
gLAB_linux -input:cfg meas.cfg -input:obs sodb2910.01o > sodb.meas
```

```
# b) Selecting satellite PRN14:
gawk '{if ($6==14) print $0}' mhcb.meas > mhcb_14.meas
gawk '{if ($6==14) print $0}' monb.meas > monb_14.meas
gawk '{if ($6==14) print $0}' sodb.meas > sodb_14.meas
```

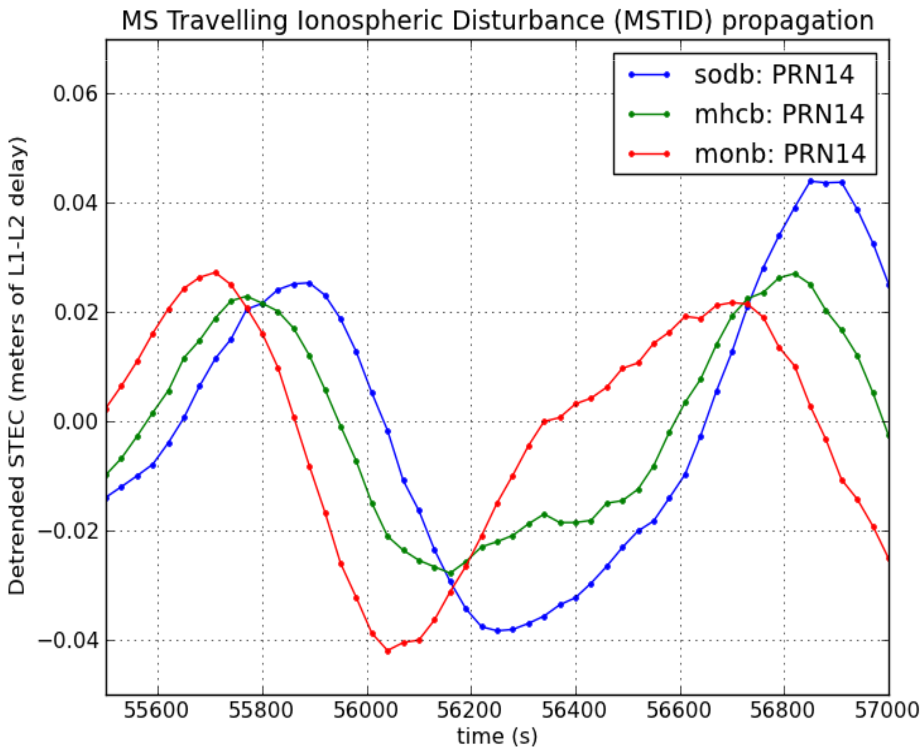


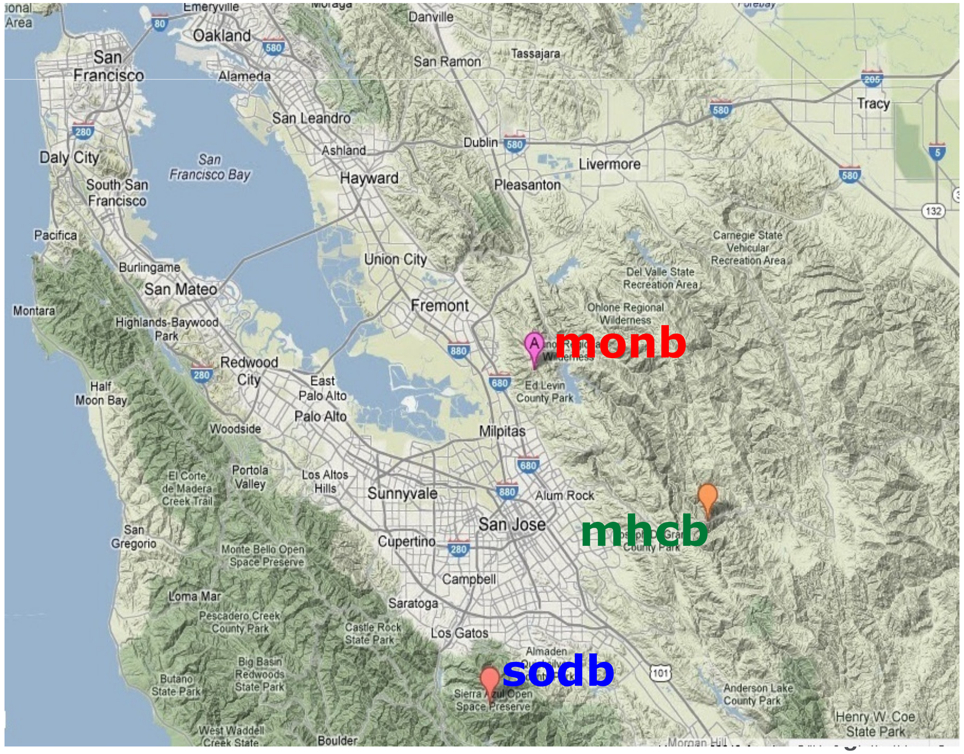

```
# c) Detrending on the geometry-free combination L1-L2:

gawk '{for (i=0;i<21;i++) {t[i]=t[i+1];l[i]=l[i+1]};t[21]=$4;l[21]=$14-$16;
      if (NR>21){tt=t[0]*t[10]*t[20];if (tt!=0) print t[10],(l[10]-(l[0]+l[20])/2)}}' mhcb_14.meas > mhcb_dLi.meas

gawk '{for (i=0;i<21;i++) {t[i]=t[i+1];l[i]=l[i+1]};t[21]=$4;l[21]=$14-$16;
      if (NR>21){tt=t[0]*t[10]*t[20];if (tt!=0) print t[10],(l[10]-(l[0]+l[20])/2)}}' monb_14.meas > monb_dLi.meas

gawk '{for (i=0;i<21;i++) {t[i]=t[i+1];l[i]=l[i+1]};t[21]=$4;l[21]=$14-$16;
      if (NR>21){tt=t[0]*t[10]*t[20];if (tt!=0) print t[10],(l[10]-(l[0]+l[20])/2)}}' sodb_14.meas > sodb_dLi.meas
```

Ex. 6: Travelling Ionospheric Disturb.

Plotting results: Execute in a single line:

```
graph.py -f sodb_dLi.meas -s.- -l "sodb: PRN14"
         -f mhcb_dLi.meas -s.- -l "mhcb: PRN14"
         -f monb_dLi.meas -s.- -l "monb: PRN14"
         --xn 55500 --xx 57000 --yn -0.05 --yx 0.07
         --xl "time (s)" --yl  "Detrended STEC (meters of L1-L2 delay)"
         -t "MS Travelling Ionospheric Disturbance (MSTID) propagation"
```





J. Sanz Subirana, J.M. Juan Zornoza, M. Hernández-Pajares

Overview

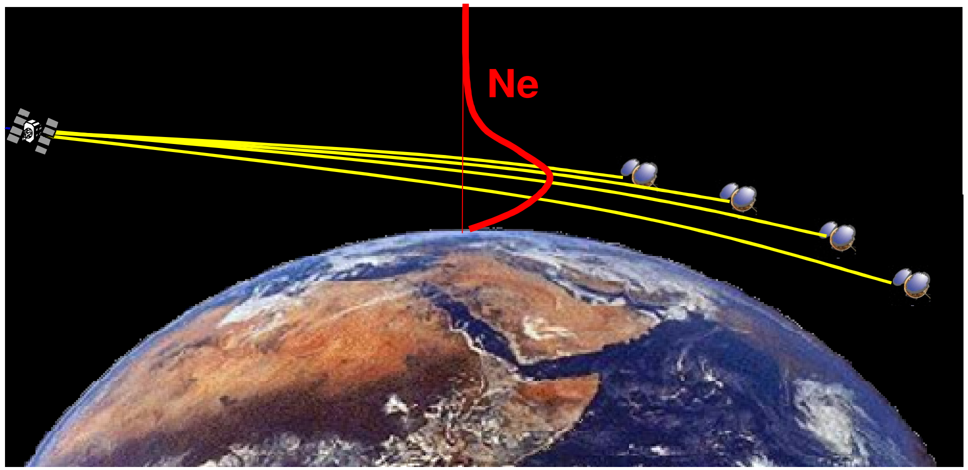
1. Introduction
2. The gLAB tool suite
3. Examples of GNSS processing using gLAB
4. Laboratory session organization

LABORATORY Session

- Starting up your laptop
- Basic: Introductory lab exercises: Iono & Posit, SF, storm,TIDs
- **Medium: Laboratory Work Projects: LWP1, LWP2**
- Advanced: Homework

LWP1:
Electron Density Profile from RO

In the previous Exercise #2 the variation of the Integrated Electron Content along the ray path (STEC) has been shown. From these integrated measurements during an occultation it is possible to retrieve the electron density profile (i.e., *Ne(h)*) of the ionosphere.

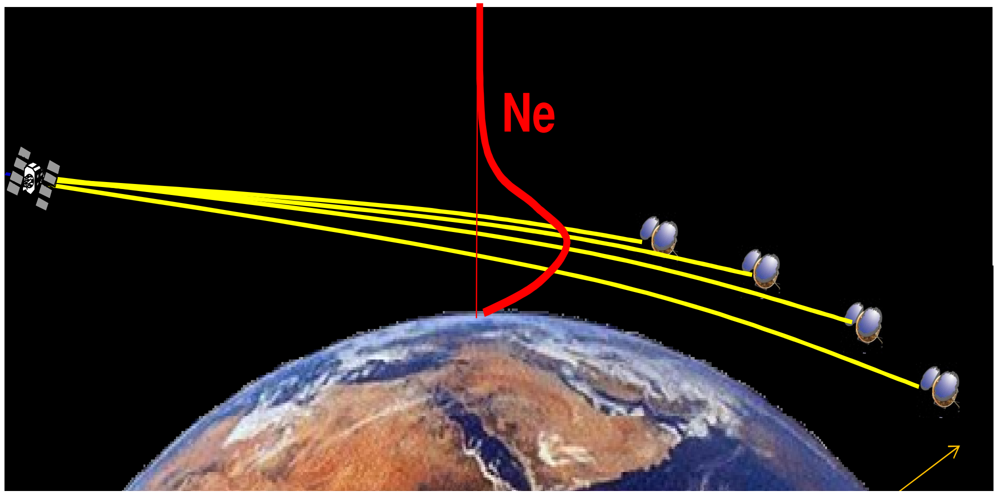


LWP1: Target

In this LWP we will retrieve the Ne(h) using a simple numerical algorithm which is equivalent to the Abel transform (see [R-6]).

LWP1:
Electron Density Profile from RO

- The basic observable of this technique is the additional delay, due to the refractivity index, of a radio signal when passing through the atmosphere.
- This additional delay is proportional to the integrated refractivity, in such a way that we can obtain an estimation of the vertical refractivity profiles using observations at different elevation angles by solving an inverse problem.
- Traditionally, the solution of this inverse problem is obtained by using the Abel inversion algorithm assuming a refractivity index that only depends on the altitude [R-5].



As it is know, *STEC* and *Ne* are related by:

$$STEC(p) = \int_{GPS}^{LEO} N_e \, dl$$

An equivalent expression, assuming spherical symmetry

$$STEC(p_n) \simeq 2\sum_{i=1}^{n} N_e(p_i)\, l_{i,n}$$

where *p* stands for the impact parameter (the closest point to the Earth centre along the optical ray path).

Tutorial associated to the **GNSS Data Processing book**
J. Sanz Subirana, J.M. Juan Zornoza, M. Hernández-Pajares

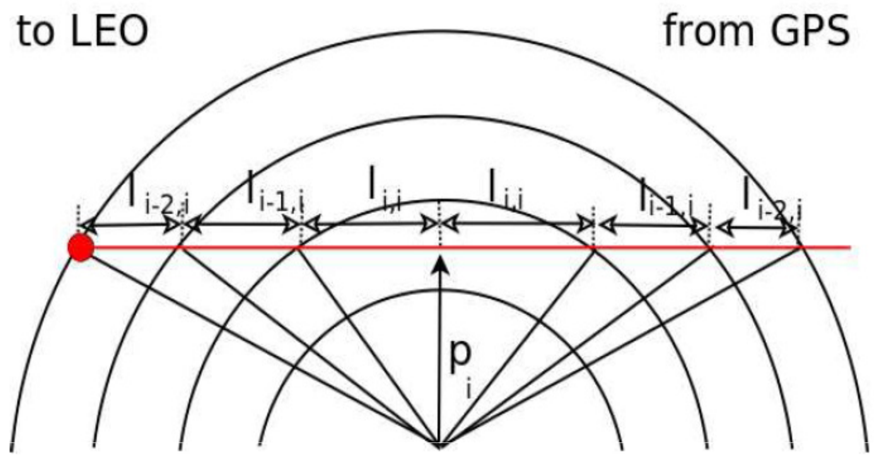
LWP1: Electron Density Profile from RO

Thence, starting from the outer ray ($p_1=r_{LEO}$), for a given ray *i*, where *i=1,…,* with impact parameter p_i, its STEC can be written in a discrete representation as:

$$STEC(p_i) = 2\, N_e(p_i)\, l_{i,i} + \sum_{j=1}^{j=i-1} N_e(p_j)\, l_{i,j}$$

where l_{ii} is the fraction of *i*th ray within the spherical layer *i*th (see [R-6]).

The previous equation defines a triangular linear equations system that can be solved recursively for the electron density *Ne(p).*



where *p* stands for the impact parameter (the closest point to the Earth centre along the optical ray path).

As measurements we use *L1-L2* carrier phases that are related with the STEC by:

$$L_1 - L_2 = \alpha\, STEC + b$$

where the bias term *b* is eliminated making differences to a reference in the arch data.

LWP1:
Electron Density Profile from RO

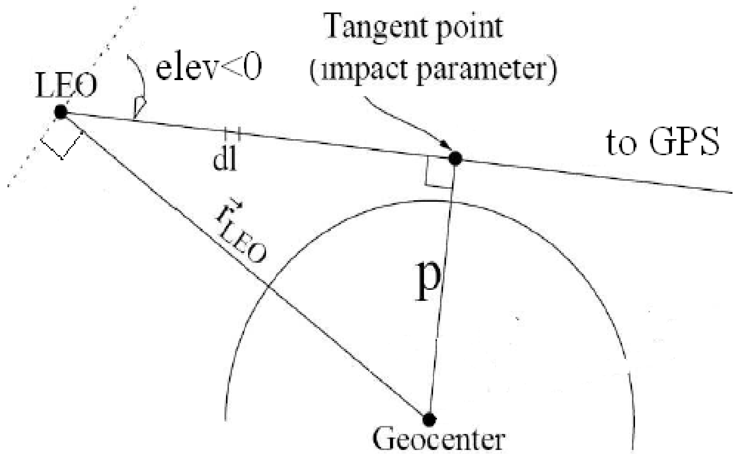
The program "**abel.perl**" implements the previous algorithm to estimate the *Ne(p)* profile from GPS L1-L2 carrier measurements.

- The input data is `[p(n),L1-L2(n)]`, with *p* in km and *L1-L2* in meters (of L1-L2 delay) where the impact parameter must be sorted from larger to lower value.
  ➔ Only measurements with negative elevation must be given (i.e., occultation).

- The output data is: `[n,p(n),L1-L2(n),Ne(n)]`, where Ne is given in e^-/m^3.

Note: the Impact parameters can be computed from the LEO elevation (elev) and its distance to Earth's centre (r_{LEO}) by (see figure):

$$p = r_{LEO}\cos(elev)$$



Tutorial associated to the **GNSS Data Processing book**
J. Sanz Subirana, J.M. Juan Zornoza, M. Hernández-Pajares

LWP1:
Electron Density Profile from RO

Exercise:

The file "**RO.obs**" contains the following fields (see Ex.5):

```
                                    |←------ LEO ------>|<------- GPS ------->|
YY DoY HH.HH CODE PRN  elev  r_LEO AR_LEO DEC_LEO  r_GPS AR_GPS DEC_GPS   L1  L2   L1-L2 arc
                      (deg)  (km) (Deg)   (Deg)    (km)  (Deg)  (Deg)   (cycles)    (m)
 1   2    3    4    5    6     7     8       9       10     11      12     13  14    15    16
```

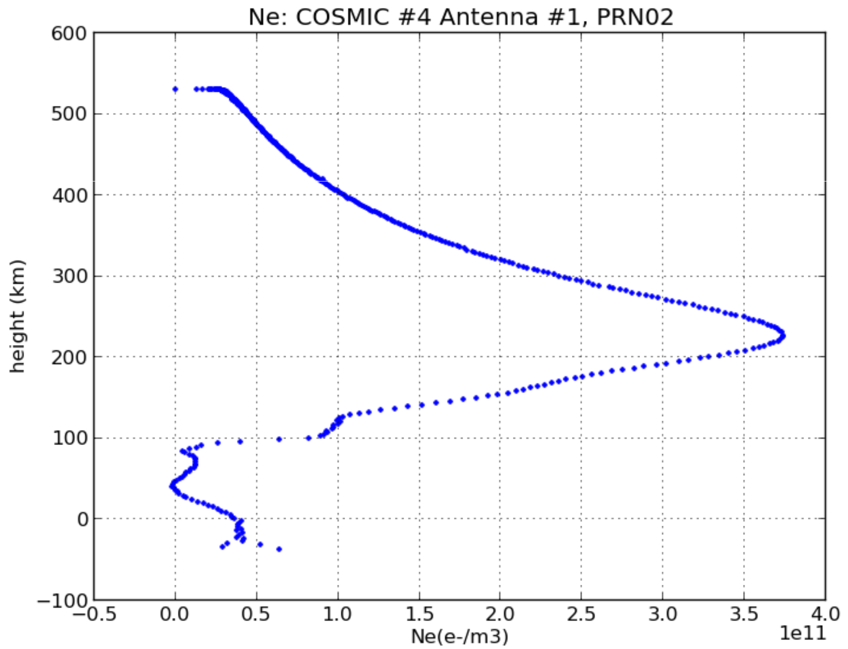
1.- Using the file **RO.obs**, select the measurements with negative elevations for GPS satellite PRN02, the **LEO #4** and **antenna #1**, and generate the input file [p(n),L1-L2(n)] for program "**abel.perl**"

```
- Selecting: CODE=l241 and PRN=02 and negative elevations (ocult)
  grep l241 RO.obs|gawk '{if ($5==02 && $6<0) print $0}'> ro.tmp

- Generating the input file
  gawk '{printf "%9.5f %7.5f \n",$7*cos($6*3.14/180),$15}' ro.tmp > abl.tmp

- Sort the file by impact parameter:
  sort -nr -k+1 abl.tmp > abl.dat
```

LWP1:
Electron Density Profile from RO

2.- Run the program "`abel.perl`" over the generated file "`abl.dat`" to compute the electron density profile Ne(p):

```
- Sort the file by impact parameter:
  cat abl.dat | abel.perl > Ne.dat
```



3.- The output file "`abl.dat`" contains the fields `[n,p(n),L1-L2(n),Ne(n)]`. Plot the electron density profile Ne as a function the impact parameter ***p*** and as a function of height above Earth.

```
- Plot1: As a function of "p"
  graph.py -f Ne.dat -x4 -y2
        --xl "Ne(e-/m3)" --yl "p (km)"

- Plot2: As a function of "height"
  graph.py -f Ne.dat -x4 -y '($2-6370)'
        --xl "Ne(e-/m3)" --yl "h (km)"
```


Tutorial associated to the **GNSS Data Processing book**
J. Sanz Subirana, J.M. Juan Zornoza, M. Hernández-Pajares

LWP1:
Electron Density Profile from RO

Questions

1. Taking into account the relationship between the electron density (N_e) and the critical frequency (f_p), i.e., minimum frequency for a signal not being reflected, [R-1]: $f_p = 8.98\sqrt{N_e} \quad (N_e \text{ in e}^-/\text{m}^3,\ f_p \text{ in Hz})$

   Compute the minimum frequency of a signal to cross through the ionosphere.

   Answer:
   From previous plot, the N_e of the maximum is 3.7E+11 e^-/m^3.
   Thence:

$$f_p = 8.98\sqrt{3.7\cdot 10^{11}} = \quad 5.46 MHz$$

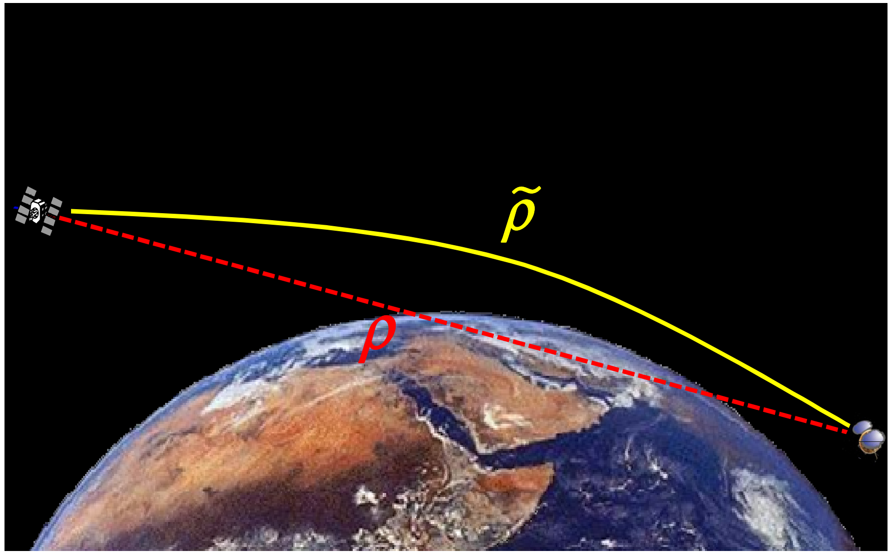
2. Calculate the height where a signal with frequency *f=4* MHz will be reflected, according to the previous plot of N_e profile.

   Answer: $N_e = (f/8.98)^2 = (4\cdot 10^6/8.98)^2 = 1.98\cdot 10^{11} \Rightarrow \sim 150\,\text{km}$

LWP2:
Atmospheric Bending in RO

LWP2: Target

This LWP is focused in depicting and analysing the effect of the atmospheric bending in Radio Occultation measurements.



A simple procedure will be given to depict the phase excess rate due to the troposphere and ionosphere over L1, L2 and LC measurements.

Note: LC is the Ionosphere Free combination of carriers L1 and L2 is given by:

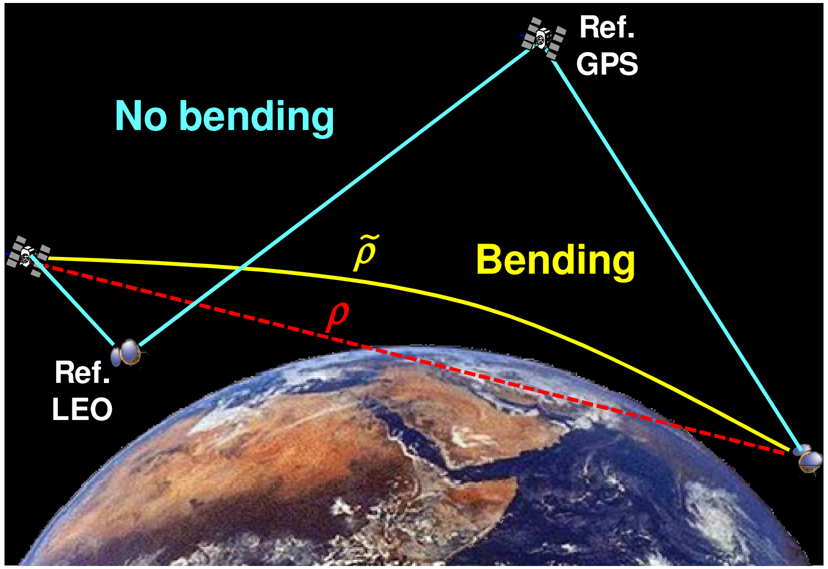
$$LC = \frac{f_1^2 L1 - f_2^2 L2}{f_1^2 - f_2^2}$$

LWP2: Atmospheric Bending in RO

Let L_1 be the carrier measurement at frequency f_1.
A procedure to depict the L1 bending is given next:

$$L_1 = \rho^* + cdt_{rec}^{sat} + B_1; \quad \rho^* = \tilde{\rho} + T - \tilde{\alpha}_1 I$$

being ρ the Euclidean distance between GPS and LEO and B a constant bias along continuous phase arc



Reference satellites (GPS, LEO) are taken in NON-OCULTATION .

- The bias B cancels, taking time differences of *L1.* Thence, from previous equation, it follows

$$\frac{\Delta\rho^*}{\Delta t} = \left(\frac{\Delta L_1}{\Delta t} - \frac{\Delta cdt_{rec}^{sat}}{\Delta t} \right)$$

- And clocks cancel taking Double-Differences between pairs of LEO and GPS satellites:

$$DD\frac{\Delta\rho^*}{\Delta t} = DD\frac{\Delta L_j}{\Delta t}$$

Assuming NO Bending in the other Sat-Sat measurements

$$\frac{\Delta\rho^*_{occult}}{\Delta t} - \frac{\Delta\rho_{occult}}{\Delta t} = DD\frac{\Delta L_j}{\Delta t} - DD\frac{\Delta\rho}{\Delta t} \neq 0$$

Note: LEO and GPS orbits are known at the level of few cm (~5cm), thus the Euclidean range can be calculated accurately and, thus, the range rate $\Delta\rho / \Delta t$

$$DD(\bullet) = \left[(\bullet)_{LEO}^{GPS} - (\bullet)_{LEO}^{GPS_{Ref}} \right] - \left[(\bullet)_{LEO_{Ref}}^{GPS} - (\bullet)_{LEO_{Ref}}^{GPS_{Ref}} \right]$$

LWP2:
Atmospheric Bending in RO

Next expressions generalize previous results for L1, L2 and Lc measurements:

$$L_j = \rho^* + cdt_{rec}^{sat} + B_j; \quad \boxed{\rho^* = \tilde{\rho} + T - \tilde{\alpha}_j I}; \quad j = 1,2$$

$$L_C = \rho^{\odot} + cdt_{rec}^{sat} + B_C; \quad \boxed{\rho^{\odot} = \hat{\rho} + T}$$

The differences in time to depict the bending

$$\frac{\Delta\rho^*}{\Delta t} = \frac{\Delta L_j}{\Delta t} - \frac{\Delta cdt_{rec}^{sat}}{\Delta t} \quad \Rightarrow DD\frac{\Delta\rho^*}{\Delta t} = DD\frac{\Delta L_j}{\Delta t}$$

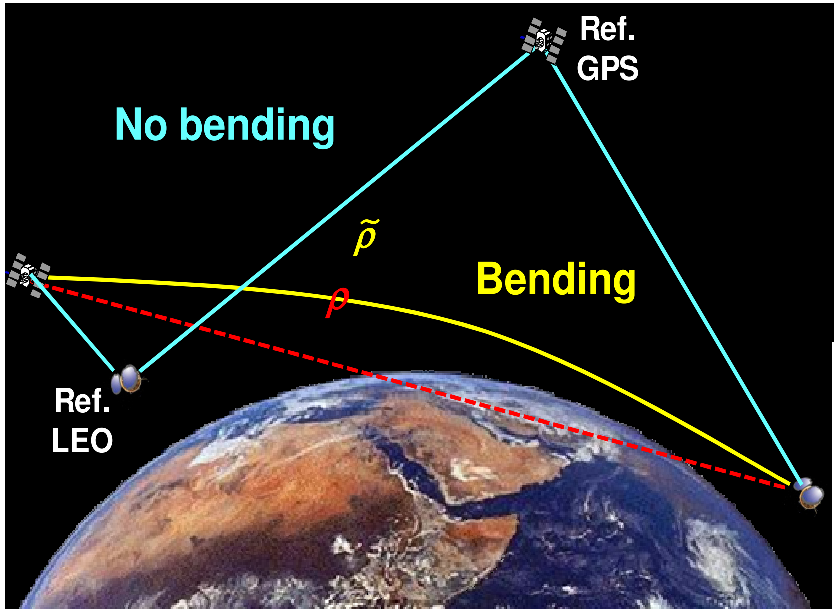
$$\frac{\Delta\rho^{\odot}}{\Delta t} = \frac{\Delta L_C}{\Delta t} - \frac{\Delta cdt_{rec}^{sat}}{\Delta t} \quad \Rightarrow DD\frac{\Delta\rho^{\odot}}{\Delta t} = DD\frac{\Delta L_C}{\Delta t}$$

And the clock term cancels taking Double Differences between LEO and GPS satellites

$$\frac{\Delta\rho^*_{occult}}{\Delta t} - \frac{\Delta\rho_{occult}}{\Delta t} = DD\frac{\Delta L_j}{\Delta t} - DD\frac{\Delta\rho}{\Delta t} \neq 0 \quad ; \ j = 1,2$$

$$\frac{\Delta\rho^{\odot}_{occult}}{\Delta t} - \frac{\Delta\rho^*_{occult}}{\Delta t} = DD\frac{\Delta L_C}{\Delta t} - DD\frac{\Delta\rho}{\Delta t} \neq 0$$

Carrier Excess Rate



Reference satellites (GPS, LEO) are taken in NON-OCULTATION .

Bending is assumed only for the GPS-LEO ray in occultation.

$$DD(\bullet) = \left[(\bullet)_{LEO}^{GPS} - (\bullet)_{LEO}^{GPS_{Ref}}\right] - \left[(\bullet)_{LEO_{Ref}}^{GPS} - (\bullet)_{LEO_{Ref}}^{GPS_{Ref}}\right]$$

Tutorial associated to the **GNSS Data Processing book**
J. Sanz Subirana, J.M. Juan Zornoza, M. Hernández-Pajares

LWP2:
Atmospheric Bending in RO

Another possibility to remove the clock term, could be to subtract the LC combination from L1 (or L2). The result will provide the “discrepancy between L1 (or L2) and LC excess ray path.

That is:

$$\left.\begin{array}{l}\dfrac{\Delta\rho^{*}_{occult}}{\Delta t}=\dfrac{\Delta L_j}{\Delta t}-\dfrac{\Delta cdt^{sat}_{rec}}{\Delta t};\; j=1,2\\[2ex]\dfrac{\Delta\rho^{\odot}_{occult}}{\Delta t}=\dfrac{\Delta L_C}{\Delta t}-\dfrac{\Delta cdt^{sat}_{rec}}{\Delta t}\end{array}\right\}\Rightarrow\dfrac{\Delta\rho^{*}_{occult}}{\Delta t}-\dfrac{\Delta\rho^{\odot}_{occult}}{\Delta t}=\dfrac{\Delta L_j}{\Delta t}-\dfrac{\Delta L_C}{\Delta t};\; j=1,2$$

Notice that the Euclidian range rate is not needed to subtract as in the previous case, because it is cancelled when taking the difference between L1 (or L2) and Lc. Other delays can also be cancelled...

In the following exercises, we will plot the previous combinations and discuss the different contribution of the ionosphere and troposphere to the phase excess rate.

LWP2:
Atmospheric Bending in RO

Exercise:

The program "**RO.perl**" uses the "RO.obs" as input data and computes the following combinations of RO measurements:

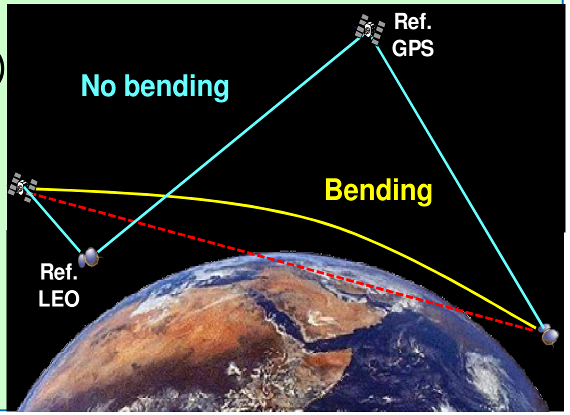
The **output file**:

```
  1    2    3   4   5    6   7   8      9         10       11     12    13   14
sec CODE PRN  p  dRho dL1 dL2 dLc d(L1-Lc) d(L2-Lc) DDdRho DDdL1 DDdL2 DDLc
                                        (units m/s)
```

where

- Rho is the Euclidean Distance between GPS and LEO
- d: means differences in time
- DD: means Double Differences between GPS and LEOs satellites.
  Note: the GPS PRN13 and LEO "l252" are used as reference satellites.
  (the rays between these satellites are not in occultation)

The results are computed for the RO between GPS PRN02 and LEO "l241" (the same occultation as in previous cases)



The aim of this exercise is to analyse the phase excess rate in the different combinations due to the bending of the ray.

LWP2:
Atmospheric Bending in RO

1.- Run the program "RO.perl" over the file "RO.obs" and generate the combinations of measurements indicated in the previous table.

Note: the results are provided for the occultation associate to **PRN=02** and "**CODE**=1241", that corresponds to **LEO=4** and **Antenna 1**.

This is hard code in the program, but can be changed, as well.

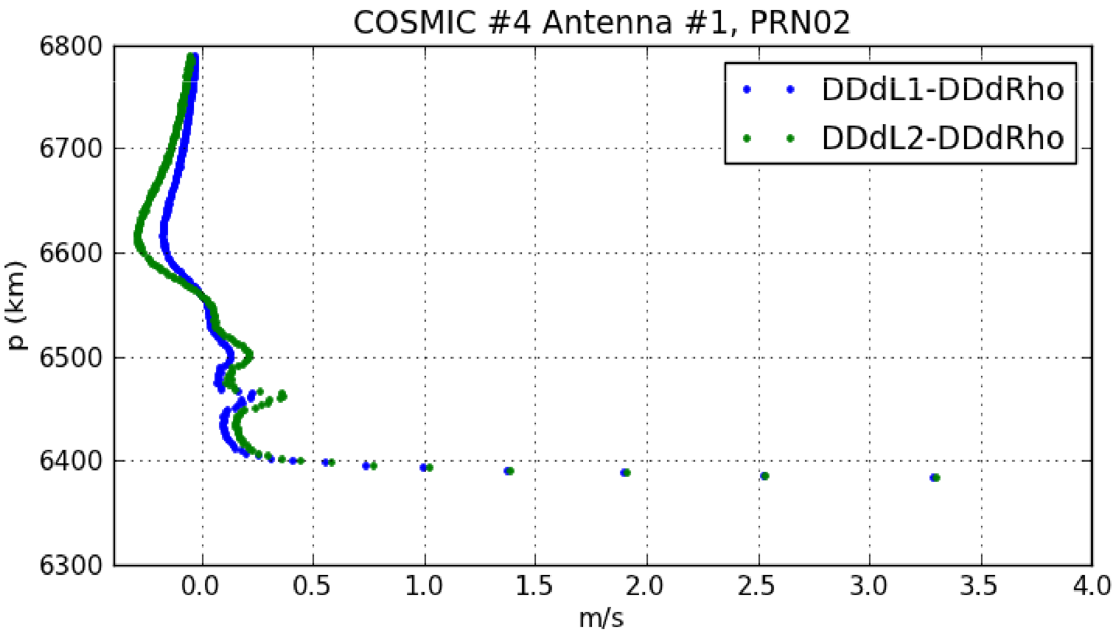
```
- Execute (the file RO.obs, must be available in the directory)

  RO.perl > bending.dat
```

2.- Discuss next plots:

LWP2: Atmospheric Bending in RO

P1.- Plot the impact parameter p as a function of "DDdL1-DDdRho" and DDdL2-DDdRho. Discuss the results found.

```
graph.py -f bending.dat -x'($12-$11)' -y4  -l "DDdL1-DDdRho"
         -f bending.dat -x'($13-$11)' -y4  -l "DDdL2-DDdRho"  --xn -0.4 --xx 4
```



Q1: Justify the discrepancy between the two plots.

Answer 1:

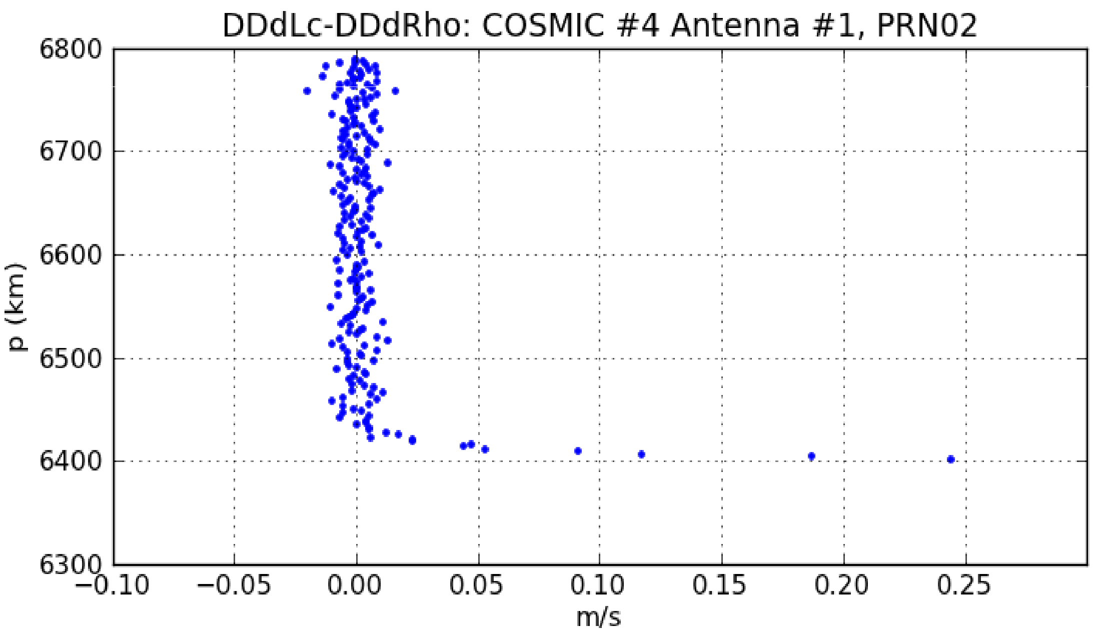
- The curves in the plot show phase excess rate due the effect of both ionosphere and troposphere.
- As the bending in the ionosphere is a frequency dependent effect, the contribution is different for each signal (L1 and L2).

Tutorial associated to the **GNSS Data Processing book**
J. Sanz Subirana, J.M. Juan Zornoza, M. Hernández-Pajares

LWP2:
Atmospheric Bending in RO

P2.- Plot the impact parameter p as a function of DDdLc-DDdRho. Discuss the results found.

```
graph.py -f bending.dat -x'($14-$11)' -y4 --xn -0.1 --xx 0.3 --xl "m_Lc/s"
          --yl "p (km)" -t"DDdLc-DDdRho: COSMIC #4 Antenna #1, PRN02"
```



Q2: Why there is no excess ray path for p>6420 km?

Answer 2:

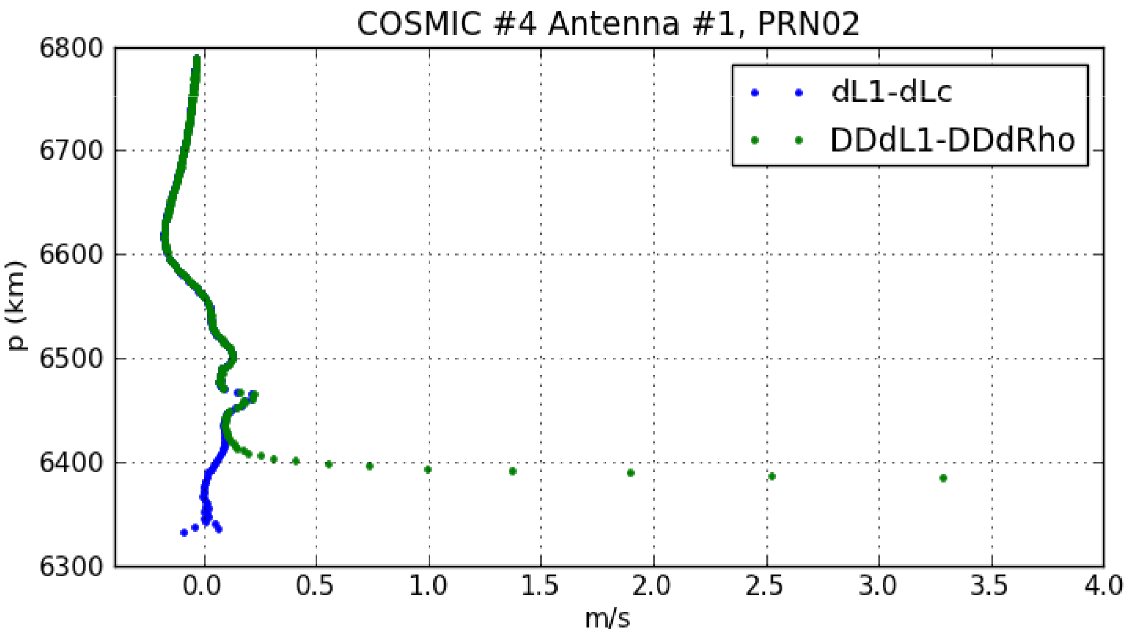
- The ionospheric bending effect on L1 and L2 is proportional to the inverse of squared frequencies (first order) and cancels in the ionosphere-free combination of carriers Lc.
- There is only bending effect in Lc due to the troposphere, which produces the path at the bottom of the figure.

Tutorial associated to the **GNSS Data Processing book**
J. Sanz Subirana, J.M. Juan Zornoza, M. Hernández-Pajares

LWP2:
Atmospheric Bending in RO

P3.- Plot the impact parameter p as a function of "dL1-dLc" and DDdL1-DDdRho. Discuss the results found.

```
graph.py -f bending.dat –x9 -y4 -l "dL1-dLc"
         -f bending.dat -x'($12-$11)' –y4 -l "DDdL1-DDdRho"
          --xl "m_L1/s" --yl "p (km)" --xn -0.4 --xx 4 -y4
```



Q3: Justify the discrepancy between the two plots.

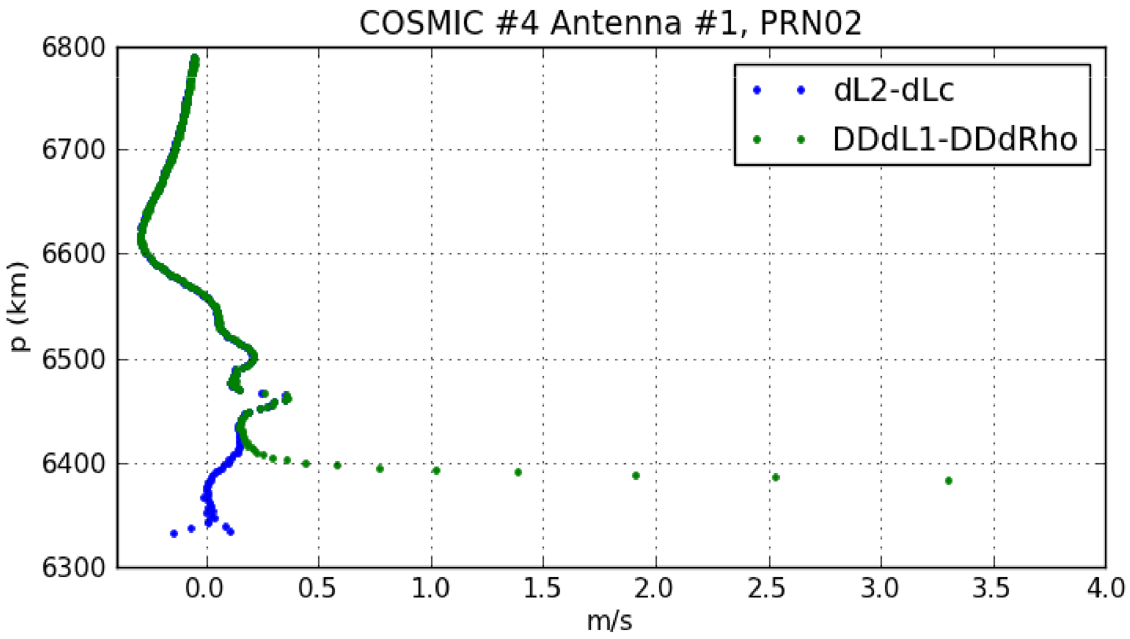
Answer 3:

- **DDdL1-DDdRho** accounts for the contribution of ionosphere and troposphere. The troposphere produces the large drift at the bottom.
- **dL1-dLc** cancels common effects in both signals, like troposphere and ionosphere. <u>But, as there is no bending effect due to the iono. on Lc</u> (see previous plot P2) , thence, <u>the curves match for *p> 6420 km.*</u>

LWP2:
Atmospheric Bending in RO

P4.- Plot the impact parameter p as a function of "dL2-dLc" and DDdL2-DDdRho. Discuss the results found.

```
graph.py -f bending.dat -x10 -y4 -l "dL2-dLc"
         -f bending.dat -x'($13-$11)' -y4 -l "DDdL1-DDdRho"
          --xl "m_L2/s" --yl "p (km)" --xn -0.4 --xx 4 -y4
```



Q4: Justify the discrepancy between the two plots.

Answer 4:

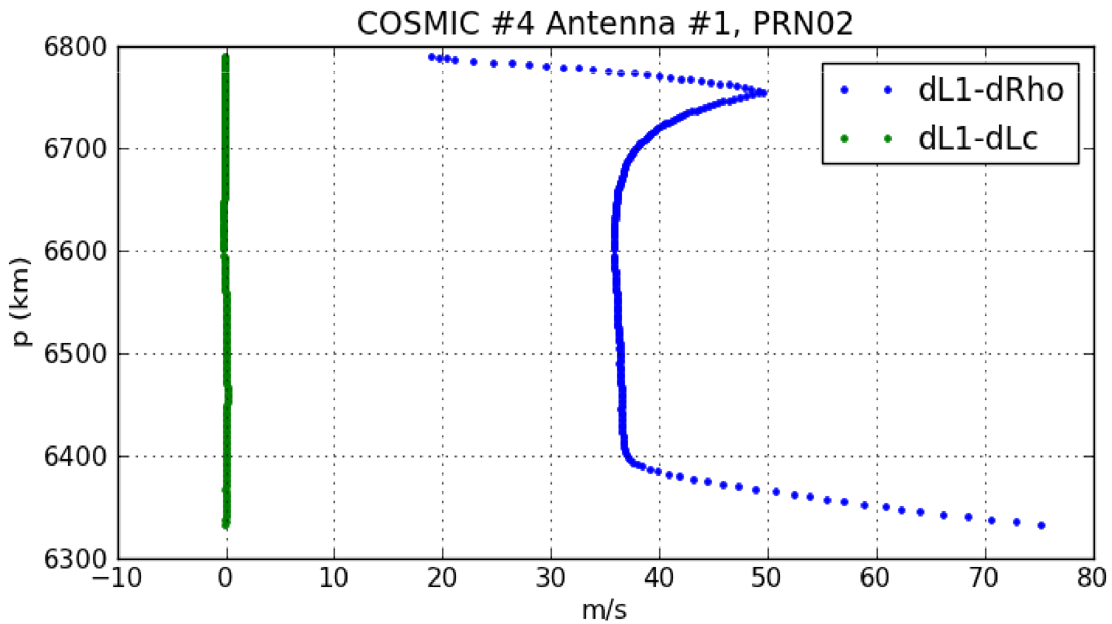
- The same answer as in previous plot.

Tutorial associated to the **GNSS Data Processing book**
J. Sanz Subirana, J.M. Juan Zornoza, M. Hernández-Pajares

LWP2: Atmospheric Bending in RO

P5.- Plot impact parameter p as a function of "dL1-dRho" and dL1-dLc. Discuss the results found.

```
graph.py -f bending.dat -x'($6-$5)' -y4 -l "dL1-dRho"
         -f bending.dat -x9 -y4 -l "dL1-dLc"
         --xl "m_L1/s" --yl "p (km)" -t"COSMIC #4 Antenna #1, PRN02"
```



Q5: Justify the discrepancy between the two plots.

Answer 5:

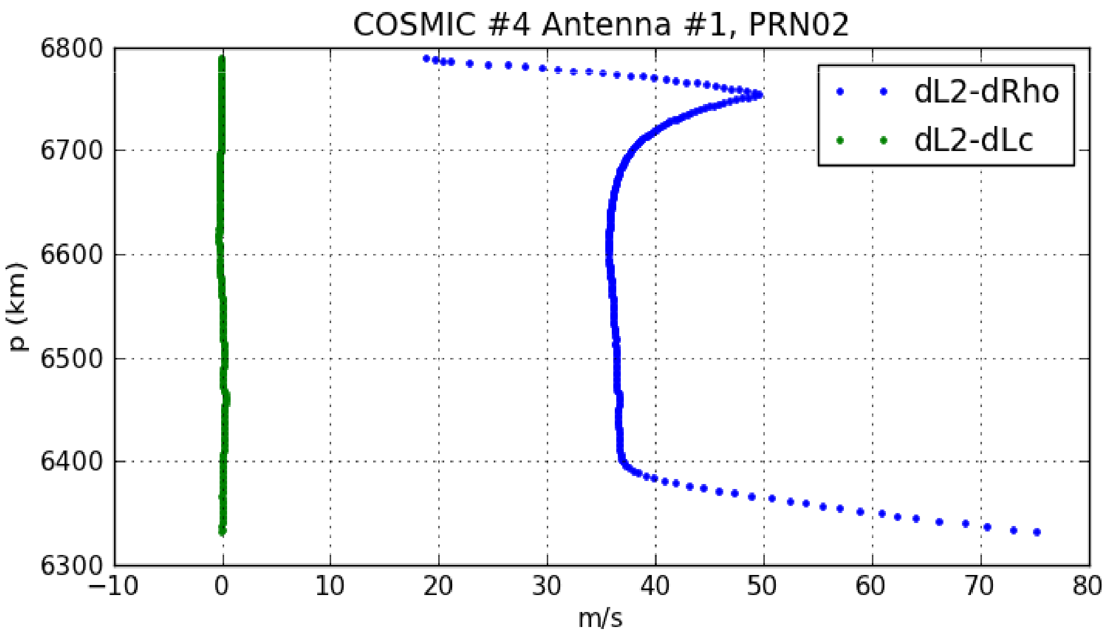
- The large drift in dL1-dRho curve is due to the satellites (GPS and LEO) clock drift.
- These large clock variations do not allow to see the atmospheric bending effect.

Tutorial associated to the **GNSS Data Processing book**
J. Sanz Subirana, J.M. Juan Zornoza, M. Hernández-Pajares

LWP2: Atmospheric Bending in RO

P6.- Plot "dL2-dRho" and dL2-dLc as a function of time.
Discuss the results found.

```
graph.py -f bending.dat -x'($7-$5)' -y4 -l "dL2-dRho"
         -f bending.dat -x10 -y4 -l "dL2-dLc"
         --xl "m_L2/s" --yl "p (km)" -t"COSMIC #4 Antenna #1, PRN02"
```



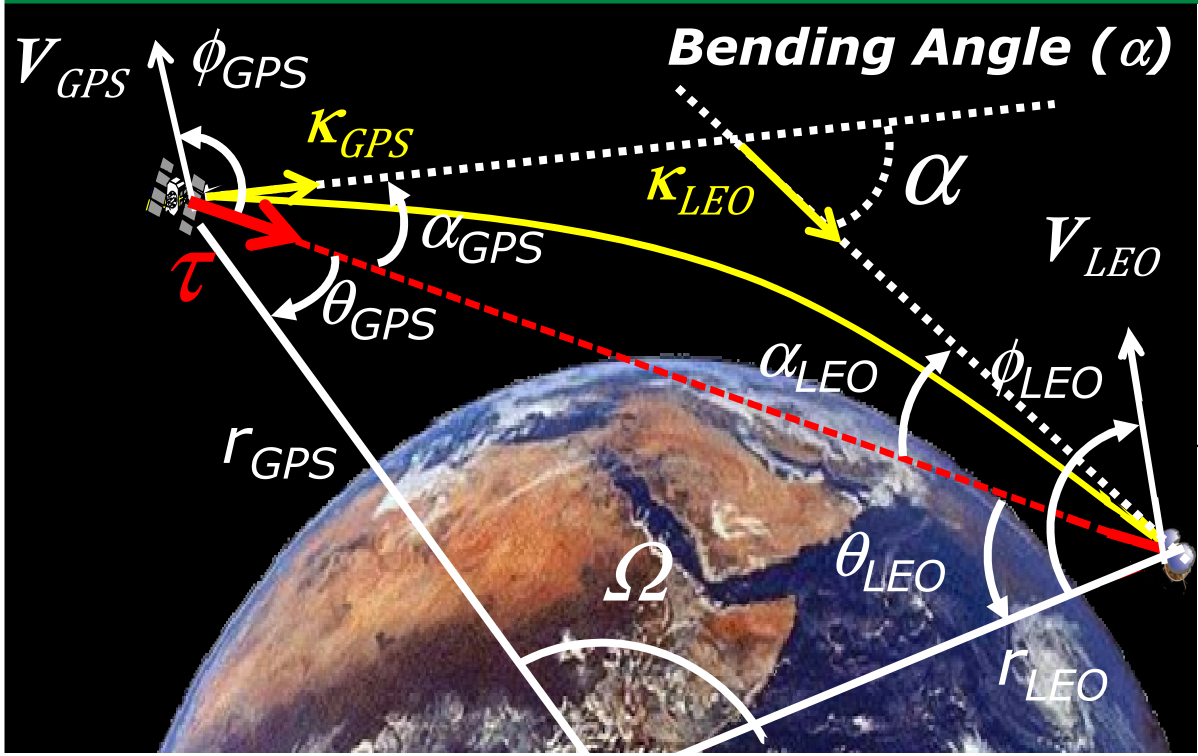
Q6: Justify the discrepancy between the two plots.

Answer 6:

- The same answer as in previous plot.

Tutorial associated to the **GNSS Data Processing book**
J. Sanz Subirana, J.M. Juan Zornoza, M. Hernández-Pajares

LWP2: Atmospheric Bending in RO



$$\lambda \Delta f = \mathbf{v}_{GPS} \cdot (\mathbf{k}_{GPS} - \boldsymbol{\tau}) - \mathbf{v}_{LEO} \cdot (\mathbf{k}_{LEO} - \boldsymbol{\tau})$$

$$n_{GPS}\ r_{GPS} \sin(\alpha_{GPS} + \theta_{GPS}) = n_{LEO}\ r_{LEO} \sin(\alpha_{LEO} + \theta_{LEO}) = a$$

$$\alpha = \alpha_{GPS} + \alpha_{LEO}$$

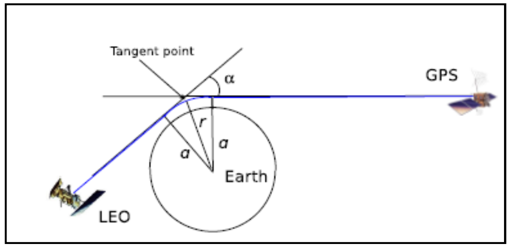
The Bending Angle $\alpha(a)$ can be derived iteratively from this equations system, see [R-6]

Derivation of bending angle $\alpha(a)$ from Excess Phase Rate (or Atmospheric Doppler Shift $\lambda \Delta f$):

$$\frac{\Delta\rho_{occult}}{\Delta t} = (\mathbf{v}_{GPS} - \mathbf{v}_{LEO}) \cdot \boldsymbol{\tau}$$

$$\frac{\Delta\rho^{*}_{occult}}{\Delta t} = \mathbf{v}_{GPS} \cdot \mathbf{k}_{GPS} - \mathbf{v}_{LEO} \cdot \mathbf{k}_{LEO}$$

$$\lambda \Delta f = \frac{\Delta\rho^{*}_{occult}}{\Delta t} - \frac{\Delta\rho_{occult}}{\Delta t} = DD\frac{\Delta L_j}{\Delta t} - DD\frac{\Delta\rho}{\Delta t}$$



Backup

LWP2:
Atmospheric Bending in RO

Comments:

From phase excess rate measurements the bending angle can be estimated.
From the bending angle, the variations of the refractivity can be computed, and from these one can then derive atmospheric quantities such as Pressure, Temperature and the partial pressure of water vapor, and electron density, among others.

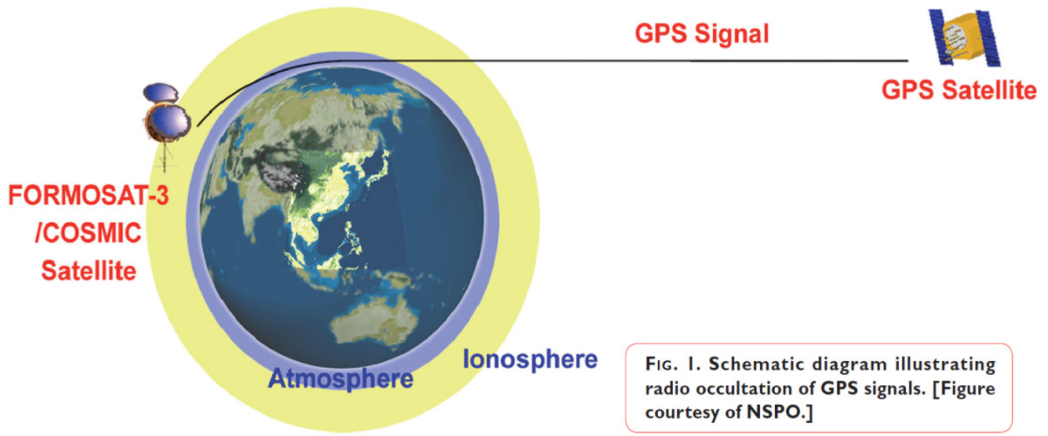


FIG. 1. Schematic diagram illustrating radio occultation of GPS signals. [Figure courtesy of NSPO.]

Overview

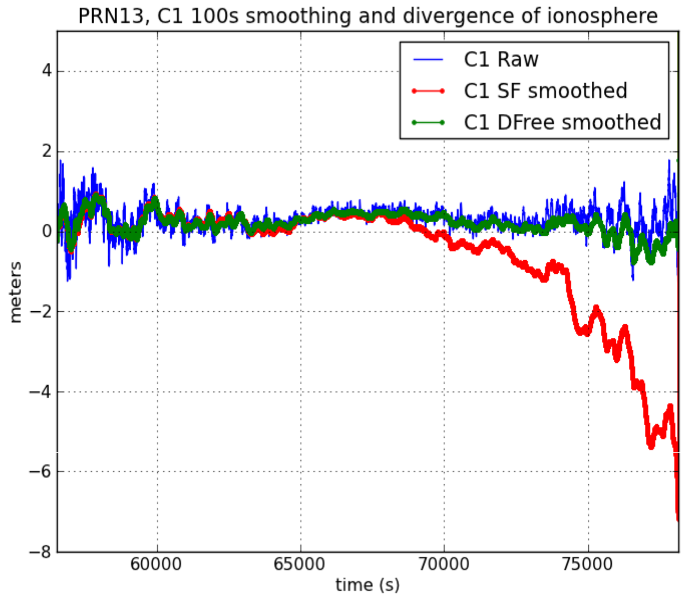
1. Introduction
2. The gLAB tool suite
3. Examples of GNSS processing using gLAB
4. Laboratory session organization

LABORATORY Session

- Starting up your laptop
- Basic: Introductory lab exercises: Iono & Posit, SF, storm,TIDs
- Medium: Laboratory Work Projects: LWP1, LWP2
- **Advanced: Homework**

HW: Iono. Divergence on Smoothing

The target of this HW is to analyze the error induced by the divergence of the ionosphere (between code and carrier) into the Single-Frequency (SF) carrier smoothed code.



The Divergence Free (Dfree) and the Ionosphere Free (IFree) smoothed codes will be compared with the SF one.

This effect will be analyzed analytically and tested with single and double frequency GPS measurements under large ionospheric gradients.

Tutorial associated to the **GNSS Data Processing book**
J. Sanz Subirana, J.M. Juan Zornoza, M. Hernández-Pajares

HW: Iono. Divergence on Smoothing

The noisy code can be smoothed with the precise (but ambiguous) carrier measurements. This carrier smoothing can be done in real-time applying the Hatch filter.

The smoothing depends on the time smoothing constant or filter length. The more the filter length is used, the more smoothed the code is, but (with single frequency measurements) a higher code-carrier divergence error is induced by the ionosphere.

This is because the ionospheric refraction has opposite sign on code and carrier, being its effect twice on the difference of code and carrier. This double ionospheric refraction is propagated forward through the filter, producing a bias.

The error induced by the code-carrier divergence of the ionosphere on the single frequency smoothed codes is assessed in this exercise for different filter lengths.

HW: Iono. Divergence on Smoothing

The noisy code P can be smoothed with the precise (but ambiguous) carrier L measurements. This carrier smoothing can be done in real-time applying the Hatch filter.

$$\hat{P}(k)=\frac{1}{n}P(k)+\frac{n-1}{n}\left(\hat{P}(k-1)+L(k)-L(k-1)\right)$$

where $[n=k \text{ if } k<N]$, and $[n=N \text{ if } k\geq N]$.

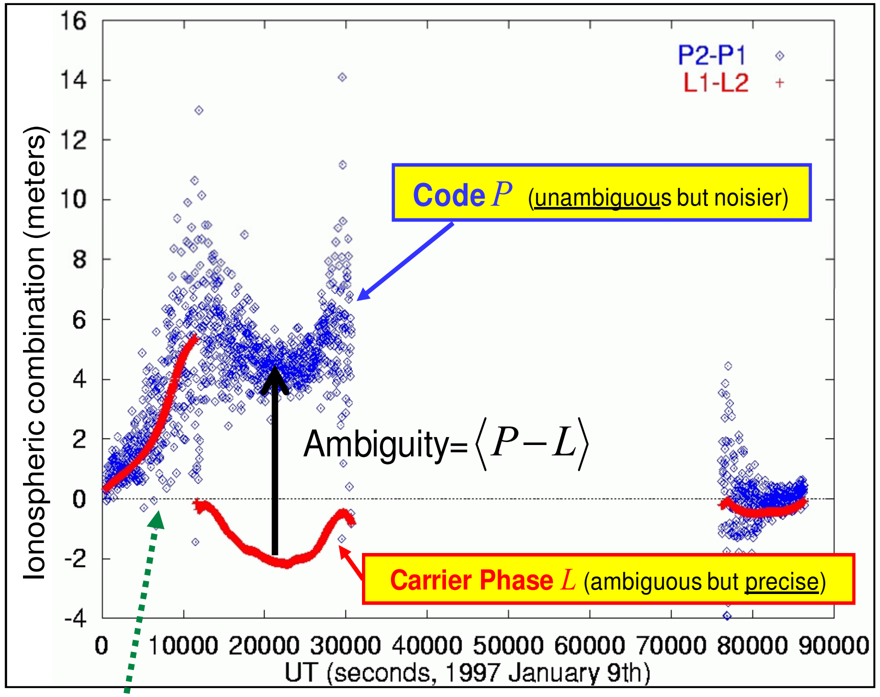
The algorithm is initialised with: $\hat{P}(1)=P(1)$.

The previous algorithm can be interpreted as real-time alignment of carrier with code:

$$\hat{P}(k)=\frac{1}{n}P(k)+\frac{n-1}{n}\left(\hat{P}(k-1)+L(k)-L(k-1)\right)$$
$$=L(k)+\langle P-L\rangle_{(k)}$$

where

$$\langle P-L\rangle_{(k)}=\frac{1}{n}\left(P(k)-L(k)\right)+\frac{n-1}{n}\langle P-L\rangle_{(k-1)}\simeq\frac{1}{n}\sum\left(P(k)-L(k)\right)$$



HW: Iono. Divergence on Smoothing

Time varying ionosphere induces a bias in the single frequency smoothed code when it is averaged in the smoothing filter.
This effect is analysed as follows:

Let:

$$P_1 = \rho + I_1 + \varepsilon_1$$
$$L_1 = \rho - I_1 + B_1 + \varsigma_1$$

Where ρ includes all non dispersive terms (geometric range, clock offsets, troposphere) and I_1 represents the frequency dependent terms (ionosphere and DCBs) . B_1 is the carrier ambiguity, which is constant along continuous carrier phase arcs and $\varepsilon_1, \varsigma_1$ account for code and carrier multipath and thermal noise.

thence,

$$P_1 - L_1 = 2I_1 - B + \varepsilon_1 \quad \Rightarrow 2I_1 : \text{Code-carrier divergence}$$

Note: the carrier noise ς_1 is neglected against code noise ε_1.

Substituting $P_1 - L_1$ in previous equation:

$$\hat{P}(k) = L(k) + \langle P - L \rangle_{(k)} = \rho(k) + I_1(k) - B_1 + \langle 2I_1 + B_1 \rangle_{(k)}$$
$$= \rho(k) + I_1(k) + \underbrace{2\left(\langle I_1 \rangle_{(k)} - I_1(k)\right)}_{bias_I}$$

$$\Rightarrow \hat{P}_1 = \rho + I_1 + bias_I + \upsilon_1$$

where υ_1 is the noise term after smoothing

where, being the ambiguity term B_1 a constant bias, thence $\langle B_1 \rangle = B_1$, and cancels in the previous expression.

Tutorial associated to the **GNSS Data Processing book**
J. Sanz Subirana, J.M. Juan Zornoza, M. Hernández-Pajares

HW: Iono. Divergence on Smoothing

Raw assessment of the **induced bias on P_1 smoothed code** by ionosphere:

- Let assume a simple model where the STEC vary linearly with time:

$$I_1(t) = I_{1_0} + I_1' \cdot t \quad \Rightarrow \quad \boxed{bias_I = 2\left(\langle I_1 \rangle_{(k)} - I_1(k)\right) = -2\tau I_1'}$$

where τ is the Hatch filter smoothing time constant (i.e., $\tau = N$ in previous eq.).

Exercise:

Proof the previous statement.

Solution:

Let be $f(t) \equiv I(t)$ *and* $y(t) \equiv \langle I \rangle_{(t)}$*. The averaging in the Hatch filter can be implemented as:*

$$y(t+\Delta T) = \frac{\tau - \Delta T}{\tau} y(t) + \frac{\Delta T}{\tau} f(t+\Delta T) \Rightarrow \frac{y(t+\Delta T) - y(t)}{\Delta T} + \frac{1}{\tau} y(t) = \frac{1}{\tau} f(t+\Delta T) \xrightarrow[\Delta T \to 0]{} \boxed{y' + \frac{1}{\tau} y = \frac{1}{\tau} f(t)}$$

Thence:

$$I_1(t) = I_{1_0} + I_1' \cdot t \quad \Rightarrow \quad \langle I_1 \rangle_{(t)} = I_1(t) - \tau I_1' \left(1 - e^{-t/\tau}\right) \Rightarrow \boxed{bias_I = 2\left(\langle I_1 \rangle_{(t)} - I_1(t)\right) \xrightarrow[t \to \infty]{} -2\tau I_1'}$$

Tutorial associated to the **GNSS Data Processing book**
J. Sanz Subirana, J.M. Juan Zornoza, M. Hernández-Pajares

HW: Iono. Divergence on Smoothing

- **Divergence Free smoothing (DFree):**

With 2-frequency measurements, the ionosphere can be removed from a combination of two carriers:

$$P_1 - L_1 - 2\alpha(L_1 - L_2) = B_{12} + \varepsilon_1 \quad \Rightarrow \quad \hat{P}_1 = \rho + I_1 + \upsilon_{12}$$

$$\alpha = \frac{f_2^2}{f_1^2 - f_2^2}$$

$$B_{12} = -B_1 - 2\alpha(B_1 - B_2)$$

➔DFree smoothed code is not affected by iono. temporal gradients, being the ionospheric delay the same as in the original code P_1 .

- **Ionosphere Free smoothing (IFree):**

Using both, code and carrier 2-frequency measurements, it is possible to remove the frequency dependent effects using the ionosphere-free combination P_C, L_C:

$$P_C = \rho + \varepsilon_C$$

$$L_C = \rho + B_C + \varsigma_C$$

$$\Rightarrow \quad \hat{P}_C = \rho + \upsilon_C$$

$$P_C = \frac{f_1^2 P_1 - f_1^2 P_2}{f_1^2 - f_2^2}$$

$$L_C = \frac{f_1^2 L_1 - f_1^2 L_2}{f_1^2 - f_2^2}$$

➔IFree smoothed code is not affected by either spatial or temporal gradients, **but is 3-times noisier** than the DFree, or the Single Freq. smoothed code.

HW: Iono. Divergence on Smoothing

1.- Multipath and measurement noise assessment on raw code measurements:

The C1 code multipath and receiver noise can be depicted using the following combination (that removes all frequency dependent and not dependent terms):

$$M_{C_1} = C_1 - L_1 - 2\alpha(L_1 - L_2)$$

$$\alpha = \frac{f_2^2}{f_1^2 - f_2^2} = \frac{1}{\gamma - 1} = 1.545 \quad ; \quad \gamma = \left(\frac{77}{60}\right)^2$$

a) Generate the "**meas**" file for PRN03:

```
[Id YY Doy sec GPS PRN el Az N. list C1C L1C C1P L1P C2P L2P]
  1  2  3   4   5   6  7  8 9   10   11   12  13  14  15 16]
```

```
gLAB_linux -input:cfg meas.cfg -input:obs UPC33510.08O|gawk '{if($6==03)print $0}'>upc3.meas
```

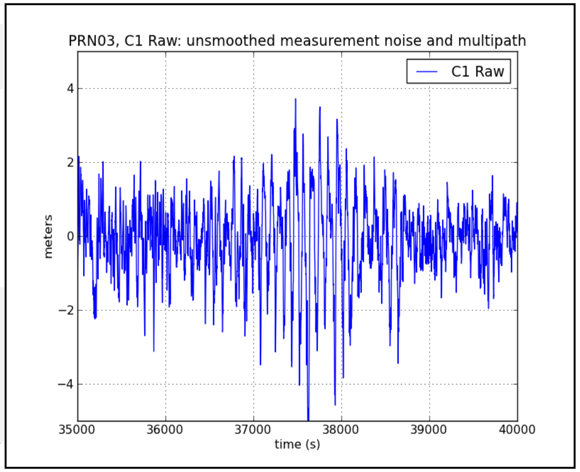
b) Using previous expression, compute the C1 multipath and code noise: :

```
gawk '{print $4,$11-$14-3.09*($14-$16)-21.3}' upc3.meas>upc3.C1
```

[*] results are Shifted by "-21.3" to remove the carrier ambiguity

c) Plot the raw (unsmoothed) measurements for PRN03:

```
graph.py -f upc3.C1 -s- -l "C1 Raw" --xn 35000 --xx 40000
         --yn -5 --yx 5  --xl "time (s)" --yl "meters"
         -t "PRN03, C1 Raw measurement noise and multipath"
```



HW: Iono. Divergence on Smoothing

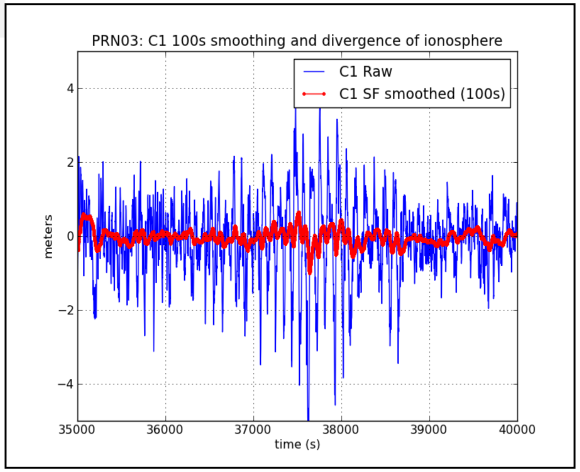
2.- Apply the Hatch filter to smooth the code using a filter length of N = 100 samples (as the measurements are at 1Hz,this means 100 seconds smoothing). Then, as in the previous case, depict the multipath and noise of the smoothed code.

a) Smoothing code (T=100sec):

```
gawk 'BEGIN{Ts=100}{if (NR>Ts){n=Ts}else{n=NR};
      C1s=$11/n+(n-1)/n*(C1s+($14-L1p));L1p=$14;
 print $4,C1s-$14-3.09*($14-$16)-21.3}' upc3.meas>upc3.C1s100
```

b) Plot results and compare with the raw C1.

```
graph.py -f upc3.C1 -s- -l "C1 Raw"
         -f upc3.C1s100 -s.- --cl r -l "C1 SF smoothed"
         --xn 35000 --xx 40000 --yn -5 --yx 5
         --xl "time (s)" --yl "meters"
         -t "PRN03: C1 100s smoothing and iono div."
```



HW: Iono. Divergence on Smoothing

3.- Remove the ionospheric refraction of C1 code and L1 carrier measurements using the following expressions to compute the Divergence Free smoothed code:

$$C_{1DFree} = C_1 - \alpha(L_1 - L_2)$$
$$L_{1DFree} = L_1 + \alpha(L_1 - L_2)$$

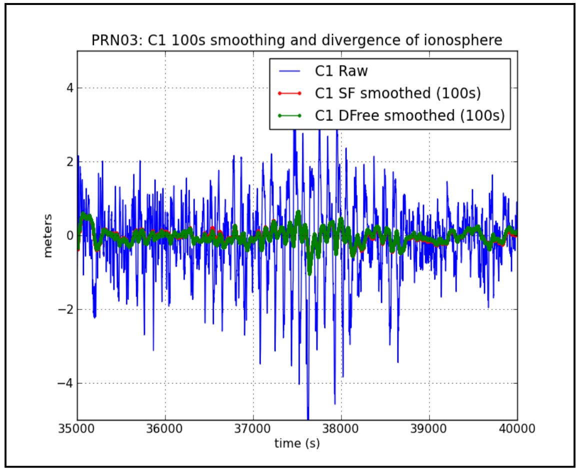
$$\alpha = \frac{f_2^2}{f_1^2 - f_2^2} = \frac{1}{\gamma - 1} = 1.545 \quad ; \quad \gamma = \left(\frac{77}{60}\right)^2$$

a) Apply the Hatch filter to compute the DFree smoothed code

```
gawk 'BEGIN{Ts=100}{if (NR>Ts){n=Ts}else{n=NR};
       C1f=$11-1.55*($14-$16);L1f=$14+1.55*($14-$16);
       C1fs=C1f/n+(n-1)/n*(C1fs+(L1f-L1p));L1p=L1f;
       print $4,C1fs-L1f-21.3}' upc3.meas  > upc3.C1DFs100
```

b) Plot results and compare with the raw C1 code:

```
graph.py -f upc3.C1 -s- -l "C1 Raw"
  -f upc3.C1s100 -s.- --cl r -l "C1 SF smoothed (100s)"
  -f upc3.C1DFs100 -s.- --cl g -l "C1 DFree smooth(100s)"
        --xn 35000 --xx 40000 --yn -5 --yx 5
        --xl "time (s)" --yl "meters"
```



HW: Iono. Divergence on Smoothing

4.- Generate the ionosphere-free combinations of code and carrier measurements to compute the Ionosphere Free (IFree) smoothed code:

$$C_{IFree} \equiv P_C = \frac{\gamma P_1 - P_2}{\gamma - 1} \quad ; \quad L_{IFree} \equiv L_C = \frac{\gamma L_1 - L_2}{\gamma - 1} \qquad \gamma = \left(\frac{77}{60}\right)^2$$

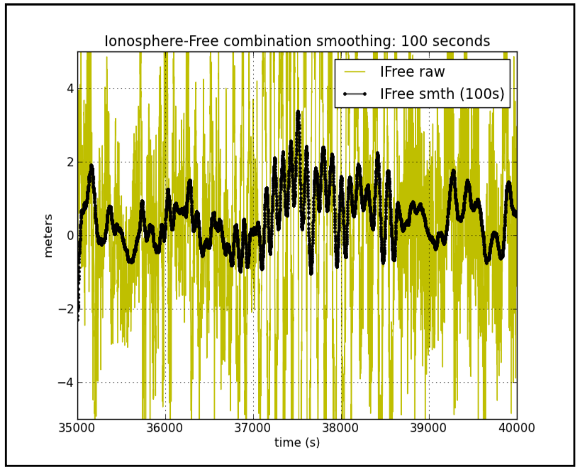
```
gawk 'BEGIN{g=(77/60)**2}{pc=(g*$13-$15)/(g-1); lc=(g*$14-$16)/(g-1);
                                       print  $4,pc-lc-3.5}' upc3.meas > upc3.PC
```

a) Apply the Hatch filter to compute the IFree smoothed code

```
gawk 'BEGIN{g=(77/60)**2}{pc=(g*$13-$15)/(g-1);
     lc=(g*$14-$16)/(g-1); if (NR>100){n=100}else{n=NR};
                  ps=1/n*pc+((n-1)/n*(ps+lc-lcp)); lcp=lc;
            print $4,ps-lc-3.5}' upc3.meas > upc3.PCs100
```

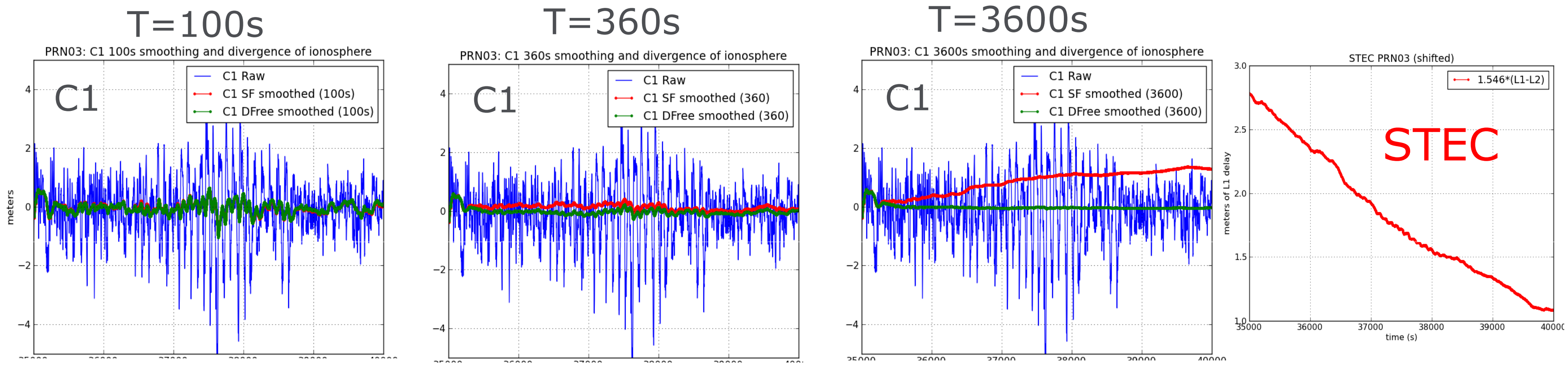
b) Plot results and compare with the unsmoothed PC:

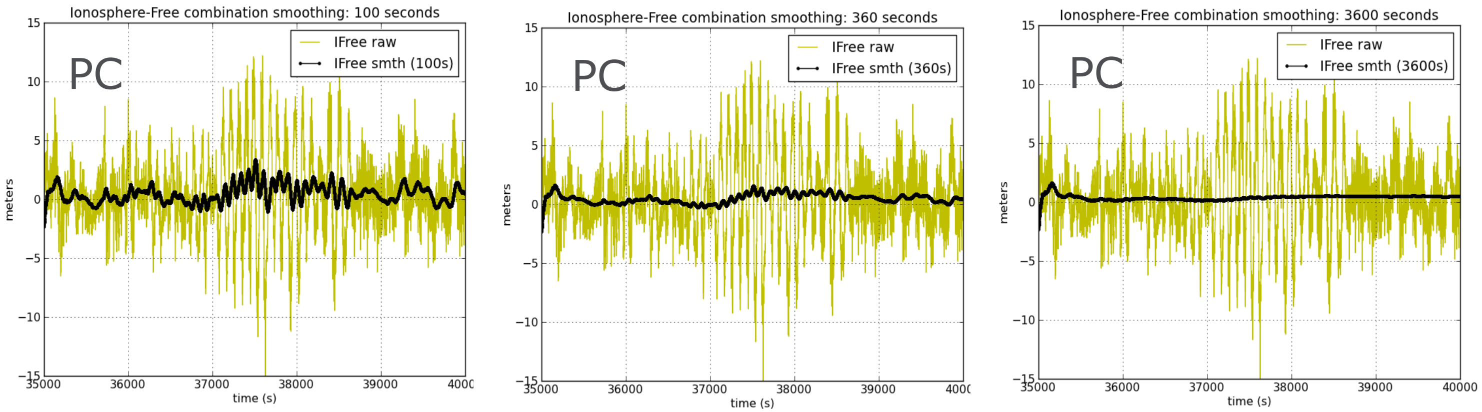
```
graph.py -f upc3.PC -s- -l "IFree raw" --cl yellow
         -f upc3.PCs100 -s.- --cl black -l "Ifree(100s)"
         --xn 35000 --xx 40000 --yn -5 --yx 5
         --xl "time (s)" --yl "meters"
```



HW: Iono. Divergence on Smoothing

5.- Repeat previous plots but using: *N=360*, *N=3600* and compare results. Plot also the ionospheric delay (from L1-L2) (see more details in [R-1]):

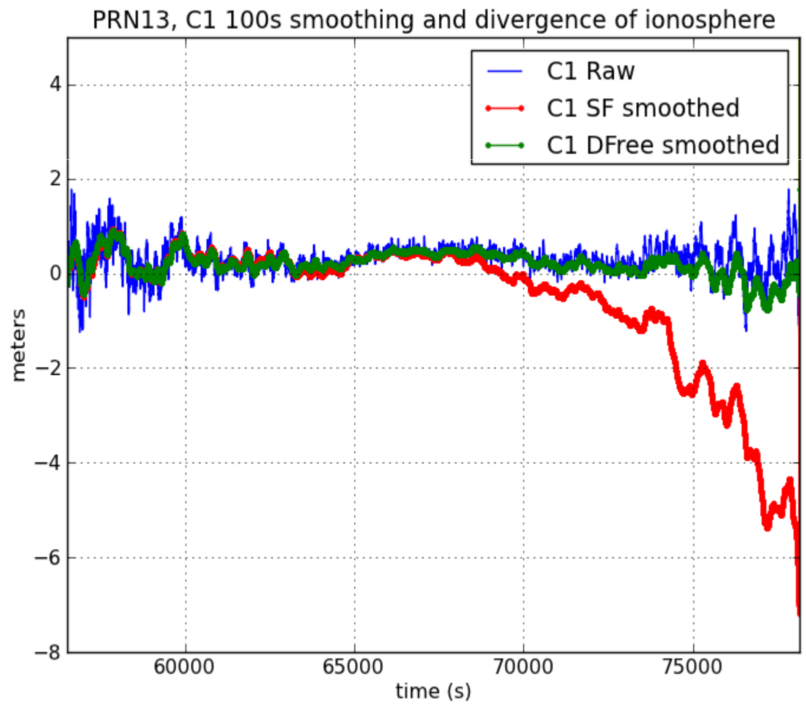


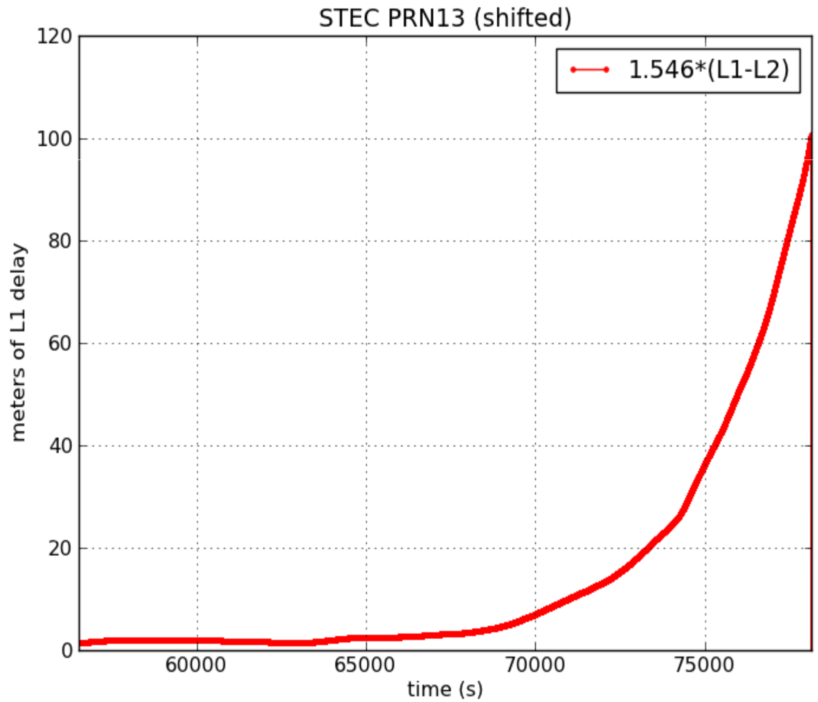


Note that the y-range in bottom row plots is 3 times larger than in top plots

HW: Iono. Divergence on Smoothing

Repeat the previous exercise using the RINEX file `amc23030.03o_1Hz` collected for the **station `amc2` during the Halloween storm**. Take *N=100* (i.e., filter smoothing time constant ***τ=100 sec***).





Bibliography

- R-1: J. Sanz-Subirana, J.M. Jaun-Zoroza, M. Hernández-Pajares. GNSS Data Processing. Volume-I and Volume-II. ESA Publications Division (in Press).
- R-2: J. Sanz-Subirana, J.M. Jaun-Zoroza, M. Hernández-Pajares. Tutorial on GNSS Data Processing Laboratory Exercises. ESA International Summer School on GNSS, 2010.
- R-3: Hernández-Pajares M., J.M. Juan, J. Sanz, "EGNOS Test Bed Ionospheric Corrections Under the October and November 2003 Storms", IEEE Transactions on Geoscience and Remote Sensing, Vol.43(10), pp.2283-2293, 2005.
- R-4: Hernández-Pajares M., J.M. Juan, J. Sanz, "Medium Scale Traveling Disturbances Affecting GPS Measurements: Spatial and Temporal Analysis", Journal of Geophysical Research, Space Physics, Vol.111, A07-S11, 2006, (doi:10.1029/2005JA011471).
- R-5: Hernández-Pajares M., J.M. Juan, J. Sanz, "Improving the Abel inversion by adding ground data LEO to radio occultations in the ionospheric sounding", Geophysical Research Letters Vol. 27(16), pp. 2743-2746, 2000.
- R-6: Hajj GA, Kursinsky ER, Romans, LJ, Bertier WI, Leroy SS. " A technical description of atmospheric sounding by GPS occultation". JASTP, Vol. 64, pp 451-469, 2002.

Acknowledgements

- The ESA/UPC GNSS-Lab Tool suite (gLAB) has been developed under the ESA Education Office contract N. P1081434.
- The other data files used in this study were acquired as part of NASA's Earth Science Data Systems and archived and distributed by the Crustal Dynamics Data Information System (CDDIS). To the NSPO/UCAR Agency for the FORMOSAT-3/COSMIC data.
- To Adrià Rovira-Garcia for his contribution to the editing of this material and gLAB updating and integrating this learning material into the GLUE.

Thank you for your attention.
Questions?

Contact: jaume.sanz@upc.edu