

Final Degree Project in Telecommunications engineering

Implementation of GNSS ionospheric models in gLAB

Author:

Deimos Ibáñez Segura

Department of Applied Mathematics IV

Universitat Politècnica de Catalunya (UPC)

Year: 2014

Director:

Prof. José Miguel Juan Zornoza
Prof. Jaume Sanz Subirana

Tutor:

Adrià Rovira Garcia



Resum

L'objectiu d'aquest Projecte Final de Carrera és incorporar els models ionosfèrics disponibles per a GNSS (Global Navigation Satellite System o Sistema Global de Navegació per Satèl·lit) en el programa gLAB (veure capítol 2). Aquesta actualització era necessària ja que han aparegut nous models ionosfèrics ja acabats o d'altres que estan en la fase final de proves. Alguns d'ells s'han desenvolupat per a noves constel·lacions de satèl·lits (com el model NeQuick per a Galileo -el sistema global de navegació per satèl·lit europeu-), altres s'han creat per a millorar la precisió (com el model F-PPP -Fast Precise Point Positioning o Posicionament de Punts Precís i Ràpid-). En qualsevol cas, actualitzar gLAB farà que el programa tingui els models de darrera generació en processament de GNSS.

Els models ionosfèrics són utilitzats durant la navegació per satèl·lit, per tant es donarà una introducció tant per a la ionosfera (veure capítol 3), com per a la navegació per satèl·lit (veure capítol 1). L'objectiu d'aquestes introduccions és donar una guia bàsica dels principis dels models ionosfèrics, accessible fins i tot per als lectors que no estiguin familiaritzats amb GNSS.

Resumen

El objetivo de este Proyecto Final de Carrera es incorporar los modelos ionosféricos disponibles para GNSS (Global Navigation Satellite System o Sistema Global de Navegación por Satélite) en el programa gLAB (ver capítulo 2). Esta actualización era necesaria ya que han aparecido nuevos modelos ionosféricos ya acabados u otros que están en la fase final de pruebas. Algunos de ellos se han desarrollado para nuevas constelaciones de satélites (como el modelo NeQuick para Galileo -el sistema global de navegación por satélite europeo-), otros se han creado para mejorar la precisión (como el modelo F-PPP -Fast Precise Point Positioning o Posicionamiento de Puntos Preciso y Rápido). En cualquier caso, actualizar gLAB hará que el programa tenga los modelos de última generación en procesamiento de GNSS.

Los modelos ionosféricos son utilizados durante la navegación por satélite, por tanto se hará una introducción tanto para la ionosfera (ver capítulo 3), como para la navegación por satélite (ver capítulo 1). El objetivo de estas introducciones es dar una guía básica de los principios de los modelos ionosféricos, accesible incluso a lectores no familiarizados con GNSS.

Abstract

The objective of this Final Degree Project is to incorporate the ionospheric models available for Global Navigation Satellite System (GNSS) (Global Navigation Satellite System) in gLAB software (see chapter 2). This update was necessary due to the appearance of new ionospheric models already finished or others which are in the final testing phase. Some of them have been developed for new satellite constellations (such as NeQuick model for European Global Navigation Satellite System (Galileo) -the european global navigation satellite system-), others have been created for improving accuracy (like Fast Precise Point Positioning (F-PPP) model -Fast Precise Point Positioning-). In any case, updating gLAB will make the program to be in the state-of-art in GNSS processing.

Ionospheric models are used during satellite navigation, so a proper introduction will be made for both the ionosphere (see chapter 3) and the satellite navigation (see chapter 1). The objective of these introductions are to give a basic guide for the principles of ionospheric models in satellite navigation, accesible even for readers not familiarized with GNSS.

Contents

Resum	III
Resumen	V
Abstract	VII
List of Figures	XIII
List of Tables	XVII
Acronyms	XIX
1 Introduction	1
2 gLAB description	5
2.1 Main gLAB features	6
2.2 Graphical User Interface (GUI)	7
2.2.1 Positioning tab	8
2.2.2 Analysis tab	9
2.3 Data Processing Core (DPC)	10
2.4 gLAB positioning example	11
3 Ionospheric models	17

3.1	Two frequency ionosphere models	20
3.1.1	Ionosphere-free combination	20
3.2	Single frequency ionosphere models	21
3.2.1	Klobuchar model	21
3.2.2	BeiDou model	24
3.2.3	NeQuick model	25
3.2.4	GIM (IONEX) model	26
3.2.5	F-PPP model	28
4	User interface changes and testing	29
4.1	User interface	30
4.1.1	Input tab	30
4.1.2	Modelling tab	32
4.1.3	Filter tab	33
4.2	Testing	34
5	gLAB applications	37
5.1	Klobuchar excessive period	37
5.2	Ionospheric model comparison	45
5.3	Other studies where gLAB has been used as a Data Processing Tool	49
6	Conclusion	51
A	List of days with Klobuchar discontinuity	53
B	Multi-platform arrangements	55
B.1	Windows	55
B.2	Macintosh	57

C F-PPP v0.50	59
Acknowledgments	63
Bibliography	65

List of Figures

1.1	GPS, GLONASS, Galileo and BeiDou satellites	2
1.2	GNSS architecture.	2
1.3	GPS, GLONASS, Galileo and BeiDou frequency bands	3
2.1	gLAB GUI initial screen	7
2.2	gLAB GUI initial Positioning tab	8
2.3	gLAB GUI initial Analysis tab	9
2.4	gLAB Input tab example	11
2.5	gLAB Preprocess tab example	12
2.6	gLAB Modelling tab example	13
2.7	gLAB Filter tab example	13
2.8	gLAB Output tab example	14
2.9	gLAB Analysis tab example	14
2.10	NEU positioning error	15
3.1	TEC content examples	19
3.2	Klobuchar model layout	21
3.3	Ionospheric Pierce Point (IPP), vertical and slant delay illustration	22
3.4	Typical Klobuchar values example	24
3.5	The five regions defined in NeQuick model	25

3.6 Global Ionospheric Maps (GIM) four point interpolation algorithm definition	27
4.1 IONEX in gLAB input tab	30
4.2 F-PPP in gLAB input tab	31
4.3 DCB in gLAB input tab	31
4.4 Ionospheric corrections in gLAB modelling tab	32
4.5 Tropospheric corrections in gLAB modelling tab	32
4.6 P1-P2 corrections in gLAB modelling tab	33
4.7 Ionospheric RMSE in gLAB filter tab	33
4.8 Generic test script diagram	35
4.9 Examples of difference between gLAB and reference	36
5.1 Typical Klobuchar ionospheric corrections for all satellites	38
5.2 Abnormal Klobuchar ionospheric corrections for all satellites	38
5.3 Typical Klobuchar ionospheric correction for a single satellite	39
5.4 Abnormal Klobuchar ionospheric corrections for a single satellite	39
5.5 Klobuchar gap values during GPS lifetime in northern hemisphere	40
5.6 Klobuchar gap values during GPS lifetime in southern hemisphere	40
5.7 Klobuchar gap values during GPS lifetime in function of geomagnetic latitude	41
5.8 Corrected Klobuchar ionospheric corrections for all satellites	42
5.9 Corrected Klobuchar ionospheric correction for a single satellite	42
5.10 Positioning error in north component between nominal and corrected Klobuchar model	43
5.11 Positioning error in east component between nominal and corrected Klobuchar model	44
5.12 Positioning error in up component between nominal and corrected Klobuchar model	44

5.13 Positioning error with Klobuchar model	45
5.14 Positioning error with NeQuick model	46
5.15 Positioning error with IGS-GIM model	46
5.16 Positioning error with F-PPP model	47
5.17 F-PPP vs ionospheric models	48

List of Tables

5.1	RMS of 3D positioning error for station hofn, DoY 319, Year 2013	47
5.2	Processing time of 3D positioning error for station hofn, DoY 319, Year 2013 (full day)	47
A.1	List of days with Klobuchar discontinuity	54
A.2	Percentage of days with Klobuchar discontinuity	54

Acronyms list

ANTEX ANTenna EXchange format

ARNS Aeronautical Radio Navigation Service

ASCII American Standard Code for Information Interchange

BeiDou Big Dipper constellation in Chinese

CPU Central Processing Unit

DAT Data Analysis Tool

DCB Differential Code Bias

DoY Day of Year

DPC Data Processing Core

EGNOS European Geostationary Navigation Overlay System

ESA European Space Agency

F-PPP Fast Precise Point Positioning

FOC Full Operational Capability

gAGE Research group of Astronomy and Geomatics

Galileo European Global Navigation Satellite System

GIM Global Ionospheric Maps

gLAb GNSS-Lab Tool suite

GLONASS GLObal NAVigation Satellite System

GNSS Global Navigation Satellite System

GPS Global Positioning System

GUI Graphical User Interface

-
- IGS** International GNSS Service
IONEX IONosphere map EXchange format
IPP Ionospheric Pierce Point
NASA National Aeronautics and Space Administration
NEU North East Up
PPP Precise Point Positioning
PRN Pseudo-Random Noise
QZSS Quasi-Zenith Satellite System
RINEX Receiver INdependent EXchange format
RMS Root Mean Square
RMSE Root Mean Square Error
RNSS Radionavigation Satellite Service
SINEX Solution (Software/technique) INdependent EXchange format
SoL Safety-of-Life
SP3 Standard Product #3
SPP Standard Point Positioning
STEC Slant Total Electron Content
TEC Total Electron Content
TECU Total Electron Content Unit
UPC Technical University of Catalonia
USA United States of America
UT Universal Time

Chapter 1

Introduction

A GNSS system is a constellation of satellites orbiting Earth, continuously transmitting signals which enables a user to determine his position in three dimensions (by solving the navigation equations), referenced from the Earth's center.

The first GNSS to appear was GPS, initially focused on military use, but in 1983 it was opened for public use (with limitations). Since that date, the number of applications using GNSS for scientific, commercial and daily use has been continuously increasing, which has made that nowadays an important chunk of the global economy is dependent on GNSS. This fact has made that several countries have decided to develop their own system.

At current date, there are four GNSS systems:

- **Global Positioning System (GPS):** The USA system. The satellite deployment started at the end of the 70's decade, but until July 1995 it was not in Full Operational Capability (FOC).
- **GLObal NAVigation Satellite System (GLONASS):** The Russian system. The satellite deployment started in 1982, achieving FOC in 1995, but during only a few years due to lack of funds. In December 2011, it was again in FOC.
- **European Global Navigation Satellite System (Galileo):** The European system. It currently has only 4 orbiting satellites. It is scheduled to reach FOC by 2019-2020.
- **Big Dipper constellation in Chinese (BeiDou):** The Chinese system. It currently has 10 orbiting satellites. It is scheduled to reach FOC by 2020.

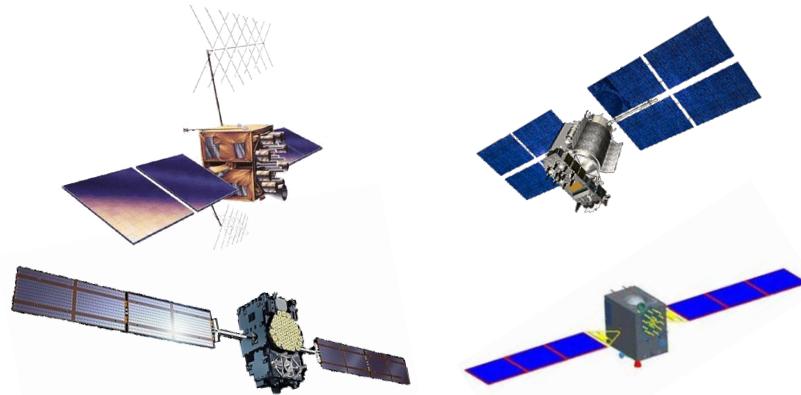


Figure 1.1: GNSS satellites: GPS IIR-M (top left), GLONASS-M (top right), Galileo IOV (bottom left) and BeiDou-M (bottom right) from [Sanz et al., 2013].

Each GNSS system is divided in three segments, as shown in figure Fig. 1.2:

- **Space Segment:** Formed by the orbiting satellites.
- **Control Segment:** Also known as ground segment, is composed of a control center for operating the system, monitoring stations distributed around the world for data collection and ground antennas for uplink data to the satellites.
- **User Segment:** Any GNSS receiver capable of determining user coordinates from the GNSS signals. For example, any modern smartphone.

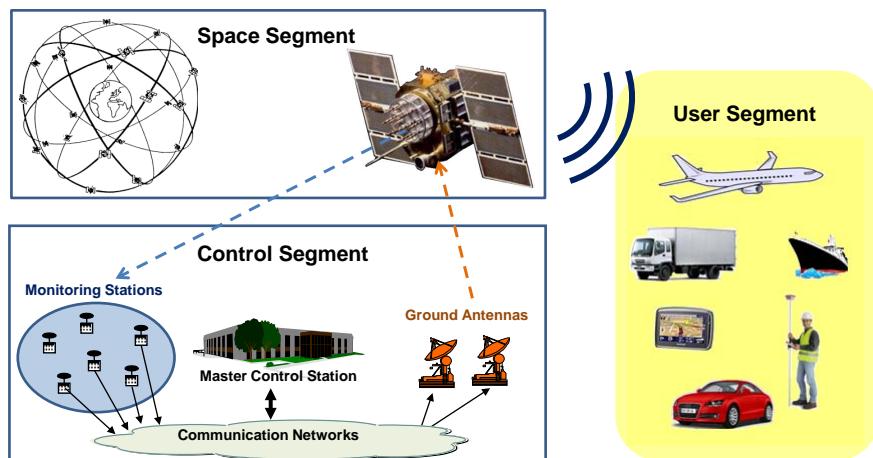


Figure 1.2: GNSS architecture from [Sanz et al., 2013].

Every GNSS system transmits signals in two or more different frequencies. There are some frequencies which are exclusive, others are shared with other GNSS systems. The signals transmitted by satellites in these frequencies can be classified in three types:

- **Carrier:** Radio frequency sinusoidal signal at a given frequency.
- **Ranging code:** Sequences of zeros and ones which allow the receiver to determine the travel time of the radio signal from the satellite to the receiver. They are called Pseudo-Random Noise (PRN) sequences or PRN codes.
- **Navigation data:** A binary-coded message providing information on the satellite ephemeris (pseudo-Keplerian elements or satellite position and velocity), clock bias parameters, satellite health status and other complementary information. This data is updated every two hours.

In Fig. 1.3 is shown the bandwidth distribution for the constellations. An exclusive bandwidth allows better signal quality due to the lack of interference, but also needs additional electronic to receive it. For the contrary, shared bandwidth allows to simultaneously receive signals from multiple constellations with a single receiver, improving accuracy if adequately combined.

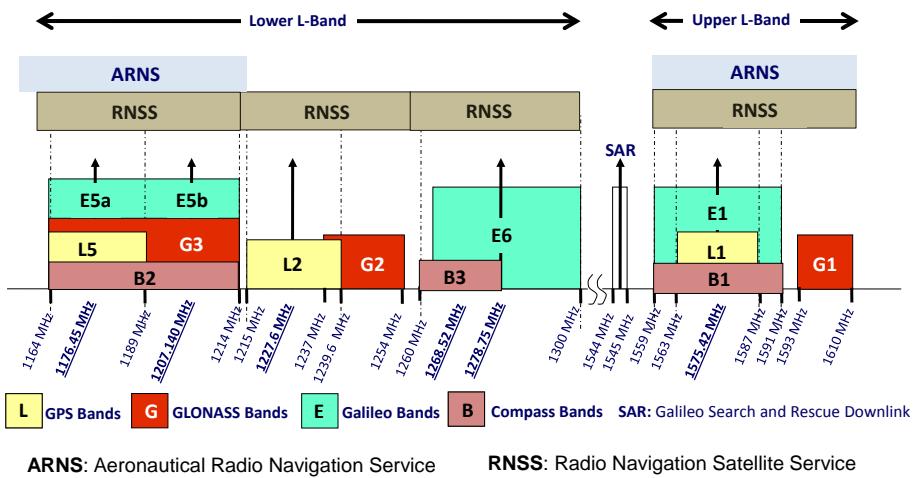


Figure 1.3: GNSS frequency bands from [Sanz et al., 2013].

As shown in Fig. 1.3, there are two frequency bands marked as Aeronautical Radio Navigation Service (ARNS), which are exclusive for GNSS signals, thus making them suitable for critical operations, such as Safety-of-Life (SoL) or aeronautical uses, hence the name. The rest of the band frequencies are for Radionavigation Satellite Service (RNSS), which are shared with radio location services (ground radars).

Furthermore, in Fig. 1.3, it is shown that:

- GPS has three frequencies, L1, L2 and L5. L1 has a civil and a military signal, L2 only has a military signal (but in the future will have a civil signal), and L5 has only a civil signal, but is still in deployment phase.
- GLONASS has three frequencies, G1, G2, G3. The three of them have both a civil and military signal, but the G3 signals is still in deployment phase)
- Galileo will have 10 civil signals in their frequencies E1, E6 and E5 (E5a and E5b signals are modulated in the same frequency E5). It will have no military signals, but some of them will have restricted access (only for public security authorities use) or for commercial services.
- BeiDou also has three frequencies, B1, B2, B3. B1 and B2 will have public signals and restricted signals (commercial or public authorities). B3 only has a restricted signal.

Finally, in order to perform GNSS positioning, the user should use the data gathered from the satellites (ranging code and navigation data) to calculate the position. As the purpose of this project is the update of gLAB, instead of giving a detailed process of the calculus to retrieve user's position, a practical example with gLAB is done in chapter 2.4.

Chapter 2

gLAB description

The GNSS-Lab Tool suite (gLAB) is an advanced interactive educational multipurpose package to process and analyze GNSS data. The first release of this software package allows full processing capability of GPS data, and partial handling of Galileo and GLONASS data.

The gLAB software tool performs a precise modelling of the GNSS observables (code and phase) at the centimeter level, allowing GPS standard and precise point positioning (SPP, PPP). gLAB also implements full processing capabilities for GPS data and is prepared to allocate future module updates, such as the expansion to Galileo and GLONASS systems, EGNOS and differential processing. It is capable of reading a variety of standard formats such as RINEX-3.00, SP3, ANTEX or SINEX files, among others. Moreover, functionality is also included for Galileo and GLONASS, being able to perform some data analysis with real multi-constellation data.

gLAB is flexible, able to run under Linux and Windows operating systems (OS) and it is provided free of charge by European Space Agency (ESA) to Universities and GNSS professionals. It is programmed in C and Python languages and is divided in three main software modules:

- **The Data Processing Core (DPC)**
- **The Data Analysis Tool (DAT)**
- **The Graphical User Interface (GUI)**

The DPC implements all the data processing algorithms and can be executed in command line. The DAT provides a plotting tool for the data analysis. The GUI consists in different graphic panels for a user friendly managing of both the DPC and DAT. Both the DPC and DAT modules may be used independently of the GUI, including them in batch files to automatically process GNSS data.

2.1 Main gLAB features

The main gLAB features are:

- **High Accuracy Positioning capability:** This software tool implements a precise modelling of the GNSS measurements (code and carrier phase) at the centimeter level, allowing both standalone GPS positioning and PPP.
- **Fully configurable:** gLAB is driven by a configuration file, where the different internal parameters are set. These range from input/output and data processing options such as Kalman filter. This ASCII configuration file can be generated from the Graphical User Interface (GUI) as well as by any experienced user, with a text editor.
- **Easy to use:** gLAB includes an intuitive GUI, with tooltips and a lot of explanations over the different options to select. Guidelines and several error and warning messages are also provided, as well as, templates and carefully chosen defaults for pre-configured processing modes.
- **Access to internal computations:** A wide tracking of internal computations is provided by gLAB through different output messages.
- **Open source:** gLAB source code is distributed under the Apache License Version 2.0. This allows the user to develop both free and commercial GNSS data processing tools using gLAB as a library.
- **Automate:** Able to be executed in command-line and to be included in batch processing.
- **Multi-platform:** gLAB can be executed in both Linux and Windows. For the former, any Linux distribution compatible with C compiler and python support will work, although Ubuntu is recommended. For the latter, the Windows versions supported are XP, 7 and 8. Furthermore, it runs in a virtual machine with any of these operative systems.

2.2 Graphical User Interface (GUI)

The GUI is an interface between the other two components (DPC and DAT). It allows the user to change parameters, and execute the other two programs with proper arguments. The initial screen can be seen in Fig. 2.1, where two main tabs may be found:

- **Positioning:** Interfaces with the DPC tool, and allows selecting the different processing options.
- **Analysis:** Interfaces with the DAT tool, and allows selecting the plotting options.



Figure 2.1: gLAB GUI initial screen.

A very important feature of gLAB are the tooltips. When the user hovers the mouse over a given option, a small box with information about the item is automatically displayed. These tooltips help the user to understand what is the effect of each option.

2.2.1 Positioning tab

The positioning tab is split into 5 different sections, which correspond to 5 different modules inside the DPC:

- **INPUT:** It is like a “driver” between the input data and the rest of the program. This module implements all the input reading capabilities and stores data into the appropriate internal structures.
- **PREPROCESS:** This module prepares the data for the next module (MODEL). It checks for cycle-slips, pseudorange-carrier phase inconsistencies and decimates the data (if required).
- **MODEL:** This module has all the functions to fully model the receiver measurements. It implements several kinds of models, which can be enabled or disabled at will.
- **FILTER:** This module implements a configurable Kalman filter, and obtains the estimations of the required parameters.
- **OUTPUT:** This module outputs the data obtained from the FILTER.

The GUI also provides two “data processing templates”, shown as buttons in the lower center part of the interface with labels “SPP Template” and “PPP Template” (see Fig. 2.2). Those “templates” automatically configure the appropriate options to carry out the desired data processing strategy.



Figure 2.2: gLAB GUI initial Positioning tab.

2.2.2 Analysis tab

The analysis tab allows configuring all the visualization options for the DAT, as shown in Fig. 2.3. There are two different areas. On the upper part, the user finds all the templates buttons. In this case, the templates are a set of preconfigured plotting options for the Graphic Details section. Clicking on any button will load all the corresponding options, allowing modifying or plotting them directly.

On the lower part, the user can configure a plot from scratch using the “Global Graphic Parameters” section below the templates. The GUI can accommodate up to four plots, (i.e. four different data series in the same graphic) although the DAT program has no plot number limitation when executed independently from the command line.

Finally, it is only to be remarked that while there are some common data that needs to be uploaded only once per graphic, user can fine tune different options of the individual plots, providing full flexibility.



Figure 2.3: gLAB GUI initial Analysis tab.

2.3 Data Processing Core (DPC)

The DPC is the processing tool of gLAB, and it has been programmed in C. It is:

- Easy to use for an advanced user.
- Modularized, in order to incorporate future updates.
- Optimized for CPU and memory usage.

The options of the DPC and GUI are basically the same with some exceptions that provide further flexibility. The DPC can be executed with the “-help” argument, which provides detailed information of the possible arguments. It is also worth mentioning that the DPC can also read the processing options from a configuration file, allowing an easy repeatability of results and automatic batch processing.

The DPC can work in four different modes:

- **Positioning Mode:** “Standard” mode, where all the processing is done and the position solution for a receiver is provided as OUTPUT messages. The minimum parameters required for this mode are an input observation file and orbit and clock products (broadcast or precise). Using precise products will also require the use of an ANTEX file.
- **Show Input Mode:** This mode only reads an input RINEX observation file and prints its measurements. No orbit nor clock products should be provided (if provided, gLAB will switch to Positioning Mode).
- **Product Comparison Mode:** This mode reads and compares two different sources of orbit and clock products. In order to use this mode, two different orbit and clock products should be provided. This mode outputs the SATDIFF, SATSTAT and SATSTATTOT messages.
- **Show Product Mode:** This mode reads a single source of orbit and clock products. In order to use this mode, a single orbit and clock product should be provided. This mode outputs SATPVT messages with the satellite coordinates and clocks.

2.4 gLAB positioning example

In this section a simple standard positioning will be done with gLAB. Although it is the least accurate, it is commonly used in mobile devices due to they only work in single frequency.

The first step is to fill in the “Input tab”. This tab is for the input data for the program.

The only necessary fields are the 'RINEX Observation File' and 'RINEX Navigation File'. The former has the raw data from the ground receiver while the latter has the satellite's ephemerides data. In this case, the observation file used will be “ebre0560.13o” (which is from the station “ebre”, day of year number 056 from year 2013) and the navigation file is “brdc0560.13n” (the broadcast file compiled by the International GNSS Service (IGS) for the same day). The interface should look like Fig. 2.4:



Figure 2.4: gLAB Input tab example.

The next step is to check the “Preprocess tab”. The purpose of this tab is to check for data consistency and validity. Some of the checks are:

- **Satellite health:** Check if the health bits in the RINEX navigation message are “0”. If not “0”, it means that a satellite is not sending valid data.
- **Cycle slip:** When using the phase of the satellite signal (not used in this example), the measurements must be continuous. When a cycle slip occurs, measurements are not continuous and they have to be reset.

It is enough to use the default configuration. More information on each options can be gathered by hovering over them until its tooltip appears. The interface should look like Fig. 2.5:



Figure 2.5: gLAB Preprocess tab example.

In the “Modelling tab”, we can select the different models to compensate the errors produced for electronic and environmental effects in the signal, such as the electronic delays or atmospheric effects.

The default options should be used. The interface should look like Fig. 2.6:



Figure 2.6: gLAB Modelling tab example.

In the “Filter tab”, represents the last step in the calculus. After checking for signal consistency and modelling error sources, a mathematic algorithm, named filter, uses all the data to calculate the current position, using also data from the previous iteration.

The default options should be used, though manipulating the filter options requires advanced knowledge. The interface should look like Fig. 2.7:



Figure 2.7: gLAB Filter tab example.

Finally, in the “Output tab”, we can select which messages are printed in the output file. gLAB can print many values from its internal calculations.

The default options are recommended. At least the OUTPUT messages should be activated. The interface should look like Fig. 2.8:



Figure 2.8: gLAB Output tab example.

Once the configuration is finished, click on the “Run gLAB” button on the bottom right hand side of the screen to start the computations. Once finished, we will switch to the ‘Analysis tab’.

In the “Analysis tab”, each of the buttons are for predefined plots. Clicking on the “NEU positioning error” button will automatically set the configuration for printing the measurement error in the three axis (North East Up (NEU)). The interface should look like Fig. 2.9:



Figure 2.9: gLAB Analysis tab example.

Clicking in the “Plot” button in bottom right hand side of the screen will show the desired plot. It should look like Fig. 2.10:

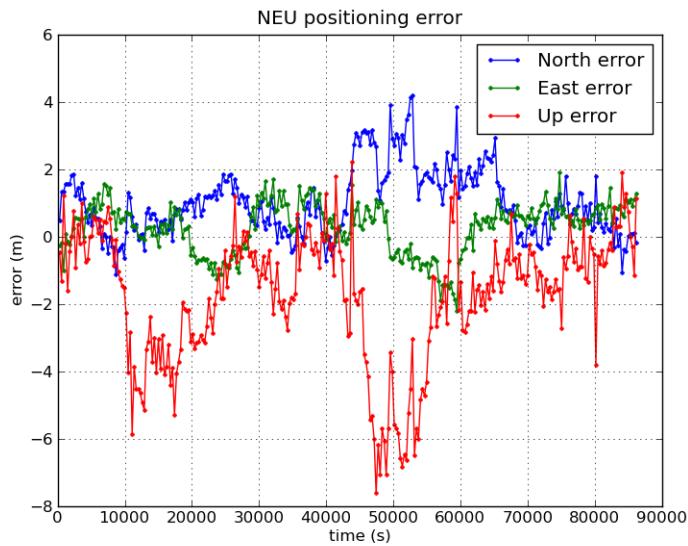


Figure 2.10: NEU positioning error.

In the plot, during most part of the day the North and East error (corresponding to a 2D positioning) is between ± 2 meters, but the Up error (height) is much more unstable (up to 8 meters). This is due to that the satellites in range tend to surround the user (if seen from user's position), there are signals from many directions, making the mean error decrease, except in the Up axis, because the Earth blocks the satellite's signals which are at the other side of the planet. Indeed, a user can see several satellites around its position in X and Y axis (which makes the errors in these axis slightly compensate each other), but not in the Z axis, because the earth blocks the satellite's signals which are at the other side of the planet.

The plot shown in Fig. 2.10 would be a typical result from standard positioning. The peak errors that appear can be to several causes, like ionospheric changes, code multipath or poor geometrics.

Chapter 3

Ionospheric models

The Earth's atmosphere is a gaseous cover surrounding the planet and is retained by the Earth gravitational field. The different regions of the atmosphere can be characterized by temperature, composition and ionization. The atmosphere changes the speed and direction of propagation of radio signals, phenomenon which is referred to as refraction. These changes affect the signal transit time, which is the basic measurement in GNSS. The variation in signal direction is generally insignificant in both layers, but the speed variations can lead up to 60 meters difference in pseudorange measurements and can not be directly measured. Therefore, it is required a mathematic model to compensate this effect. For satellite signals, which goes through atmosphere, there are two relevant layers, the troposphere and the ionosphere.

The troposphere, is a region of electrically neutral gaseous. It is found between Earth's surface and an altitude of about 60 km, and it is a non dispersive medium, meaning that the changes are independent from the frequency. A consequence of not being a dispersive medium is that the tropospheric refraction cannot be removed with dual-frequency measurements, hence being compulsory the use of a mathematic model (this is the main reason why ESA's tropospheric model has been also included in this update, see section [4.1.2](#)).

The troposphere refraction on GNSS signals appears as an extra delay in the measurement of the signal travel time. This delay is separable in two components, the hydrostatic component delay (90% of the total delay) and the wet component delay (10%). The former is caused by dry gases in the troposphere, and has a slow and predictable variation (it is the variation predicted by the tropospheric models). The latter is caused by the water vapour and condensed water in form of clouds, and therefore depends on weather conditions. This component delay is small (about tens of centimeters) and is estimated in the navigation filter only in precise navigation. No more details will be given about troposphere, though it is not the subject of study in this project.

Focusing now on the ionosphere, it is the atmosphere region extends from a height from 50 to 2000 km above the Earth. This region contains a partially ionised medium, as a result of solar X and Extreme UltraViolet (EUV) rays in the solar radiation and the incidence of charged particles, which ionize the different neutral atmospheric components.

The propagation speed of GNSS electromagnetic signals in the ionosphere depends on its electron density, which is typically driven by two main processes. During the day, the Sun's radiation ionises neutral atoms to produce free electrons and ions. During the night, the recombination process prevails, where free electrons are recombined with ions to produce neutral particles, which leads to a reduction in the electron density. The electron density (N_e) in the ionosphere changes with height, having a maximum of $N_e \simeq 10^{11}$ to $10^{12} \text{ e}^-/\text{m}^3$ around 300 – 500 km.

The ionosphere is a dispersive medium, which means that it is frequency dependent. Furthermore, dispersive mediums reflect signals lower than a threshold frequency. For the ionosphere the threshold is at 10^6 Hz , which is lower than GNSS signals, that are at the order of 1 GHz ($= 10^9 \text{ Hz}$), thus allowing satellite signals to go through.

To compute the difference between the measured range (with frequency f signal) and the Euclidean distance between the satellite and receiver, the relation is the following (from [Sanz et al., 2013]):

$$\Delta_{ph,f}^{iono} = -\frac{40.3}{f^2} \int N_e dl, \quad \Delta_{gr,f}^{iono} = +\frac{40.3}{f^2} \int N_e dl \quad (3.1)$$

The differences $\Delta_{ph,f}^{iono}$ and $\Delta_{gr,f}^{iono}$ are called the phase and code ionospheric refraction, respectively, and the integral is defined as the Slant TEC, or Slant Total Electron Content (STEC)

$$\text{STEC} = \int N_e dl \quad (3.2)$$

As it can be seen in (3.1), phase measurements are advanced on crossing the ionosphere, that is a negative delay, while the code measurements undergo a positive delay.

Usually, the STEC is given in TEC units (TECUS), where $1 \text{ TECU} = 10^{16} \text{ e}^-/\text{m}^2$ and the ionospheric delay I_f (at the frequency f) is written as

$$I_f \equiv \Delta_{gr,f}^{iono} = \alpha_f \text{ STEC} \quad (\text{units: metres of delay}) \quad (3.3)$$

with

$$\alpha_f = \frac{40.3 \cdot 10^{16}}{f^2} \text{ m}_{\text{signal delay(at frequency } f)} / \text{TECU} \quad (\text{where } f \text{ is in Hz}) \quad (3.4)$$

The TEC, and hence the ionospheric refraction, depend on the geographic location of the receiver, the hour of day and the intensity of the solar activity. Figure 3.1a (left) shows a vertical TEC map of the geographic distribution of the TEC, where the equatorial anomalies are clearly depicted around the geomagnetic equator. The figure on the right 3.1b shows the 11-year solar cycle with a solar flux plot.

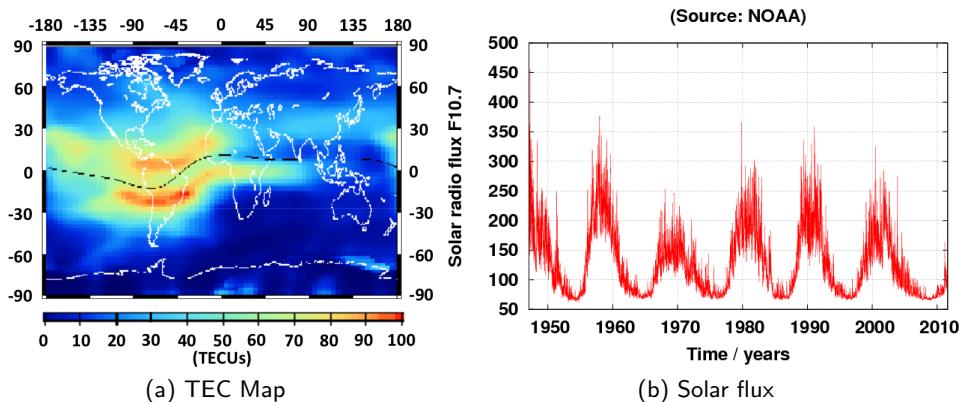


Figure 3.1: The map on the left shows the vertical Total Electron Content in TECUs at 19UT on 26 June 2000 ($1 \text{ TECU} \simeq 16 \text{ cm}$ of delay in the GPS L1 signal). The plot on the right shows the evolution of the solar flux during the last solar cycles (from [Sanz et al., 2013]).

3.1 Two frequency ionosphere models

3.1.1 Ionosphere-free combination

As explained above, ionosphere effects are frequency dependent. In concrete, first order effects depend on 99.9% on the inverse of squared signal frequency. Therefore, ionosphere effect can be eliminated through a linear combination shown in (3.5), in example the ionosphere free combination:

$$R_{iono-free} = \frac{f_1^2 R_1 - f_2^2 R_2}{f_1^2 - f_2^2} \quad (3.5)$$

Where R_1 and R_2 are the pseudorange or carrier values measured in both frequencies, f_1 and f_2 are the signal frequencies, and $R_{iono-free}$ is the resulting value without ionosphere effects.

Although it is not a mathematic model, it is by far the most effective and simplest. Nevertheless, it has two important drawbacks: it requires dual frequency receivers (many of them are single receivers to reduce costs and space) and, obviously, it requires two frequencies.

For GPS system, there is only one civilian signal in one frequency. Nevertheless, many commercial receivers are capable of tracking military signals in both frequencies, but at the cost of losing some accuracy. In the near future, it is planned to create a new civil signal in the second GPS frequency. Regarding the other GNSS system, GLONASS has two public civil signals since 2004, while Galileo and BeiDou will have them at the moment they reach Full Operational Capability (FOC).

It must be pointed out that the Precise Point Positioning (PPP) uses code and carrier phase measurements in the ionosphere-free combination to remove the ionospheric refraction, because it is one of the effects that is more difficult to model accurately.

3.2 Single frequency ionosphere models

Single frequency receivers need to apply a model to remove ionospheric effects. As mentioned above, ionosphere effects are greatly variable, hence it is necessary to adjust model parameters periodically. GPS, Galileo and BeiDou broadcast this parameters through the navigation file transmitted by the satellites, and are updated daily. GLONASS does not broadcast any ionospheric model, but any of the existing models can be applied to this constellation by applying a correction factor given by their relative squared frequency ratio.

3.2.1 Klobuchar model

Klobuchar model was designed to minimize user computational complexity and user computer storage so as to keep a minimum number of coefficients to be transmitted. It is based on an empirical approach ([[Klobuchar, 1987](#)]) and it is estimated to reduce the error by 50% (in RMS). In the navigation message, these coefficients are given in the lines labeled as “IONALPHA” and “IONBETA” in RINEX version 2 or “GPSA” and “GPSB” in version 3 in the header to broadcast its parameters.

This model is independent from user’s position (it does not apply any specific correction for any region) and the ionosphere is modeled as single layer at an altitude of 350 km. This model assumes there is a constant delay of 5 ns during night time and a half-cosine function in daytime that is centered at the 14th hour (2 pm) of local time (as shown in Fig. 3.2), whose amplitude and period are given as a function of the eight parameters broadcasted in the navigation message.

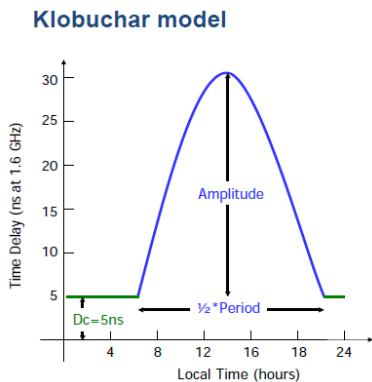


Figure 3.2: Klobuchar model layout from [[Sanz et al., 2013](#)]

To compute the slant delay, it is necessary to first calculate the Ionospheric Pierce Point (IPP), which is the intersection of the ray with the ionospheric layer at 350 km in height, then compute the vertical delay at the IPP, and finally get the slant delay by multiplying by an obliquity factor (also called the mapping function, see Fig 3.3 and equation 3.14).

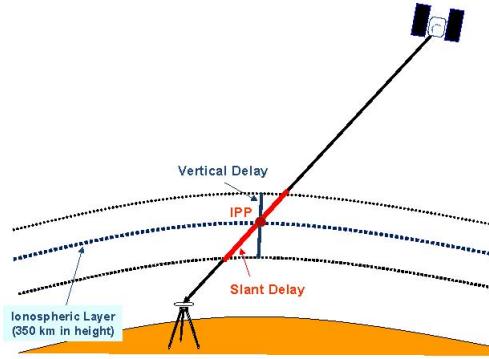


Figure 3.3: Ionospheric Pierce Point (IPP), vertical and slant delay illustration [Sanz et al., 2013]

The Klobuchar algorithm to run in a single-frequency receiver is provided as follows [Klobuchar, 1987].

Given the user's approximate geodetic latitude φ_u and longitude λ_u , the elevation angle E and azimuth A of the observed satellite and the Klobuchar coefficients α_n and β_n broadcasted in the GPS satellite navigation message:

1. Calculate the Earth-centred angle

$$\psi = \pi/2 - E - \arcsin\left(\frac{R_E}{R_E + h} \cos E\right) \quad (3.6)$$

2. Compute the latitude of the IPP

$$\phi_I = \arcsin(\sin \varphi_u \cos \psi + \cos \varphi_u \sin \psi \cos A) \quad (3.7)$$

3. Compute the longitude of the IPP

$$\lambda_I = \lambda_u + \frac{\psi \sin A}{\cos \phi_I} \quad (3.8)$$

4. Find the geomagnetic latitude of the IPP

$$\phi_m = \arcsin(\sin \phi_I \sin \phi_P + \cos \phi_I \cos \phi_P \cos(\lambda_I - \lambda_P)) \quad (3.9)$$

with $\phi_P = 78.3^\circ$, $\lambda_P = 291.0^\circ$ the coordinates of the geomagnetic pole.

5. Find the local time at the IPP

$$t = 43\,200 \lambda_I / \pi + t_{GPS} \quad (\lambda_I \text{ in radians, } t \text{ in seconds}) \quad (3.10)$$

where $0 \leq t < 86\,400$. Therefore:

If $t \geq 86\,400$, subtract 86 400. If $t < 0$, add 86 400.

6. Compute the amplitude of ionospheric delay

$$A_I = \sum_{n=0}^3 \alpha_n (\phi_m / \pi)^n \quad (\text{seconds}) \quad (3.11)$$

If $A_I < 0$, then $A_I = 0$.

7. Compute the period of ionospheric delay

$$P_I = \sum_{n=0}^3 \beta_n (\phi_m / \pi)^n \quad (\text{seconds}) \quad (3.12)$$

If $P_I < 72\,000$, then $P_I = 72\,000$.

8. Compute the phase of ionospheric delay

$$X_I = \frac{2\pi(t - 50\,400)}{P_I} \quad (\text{radians}) \quad (3.13)$$

9. Compute the slant factor (ionospheric mapping function)

$$F = \left[1 - \left(\frac{R_E}{R_E + h} \cos E \right)^2 \right]^{-1/2} \quad (3.14)$$

10. Compute the ionospheric time delay

$$I_1 = \begin{cases} [5 \cdot 10^{-9} + A_I \cos X_I] \times F, & |X_I| < \pi/2 \\ 5 \cdot 10^{-9} \times F, & |X_I| \geq \pi/2 \end{cases} \quad (3.15)$$

The delay I_1 is given in seconds and is referred to the GPS L1.

Although this algorithm is provided to estimate the ionospheric delay in the GPS, it can also be used to estimate the ionospheric time delay in other frequency signals or for the GLONASS and Galileo signals, as well. Indeed, taking into account that the ionospheric delay is inversely proportional to the square of the signal frequency, the delay for any GNSS signal transmitted on frequency f_k is given by

$$I_k = \left(\frac{f_1}{f_k} \right)^2 I_1 \quad (3.16)$$

A typical result of Klobuchar values is shown in Fig 3.4

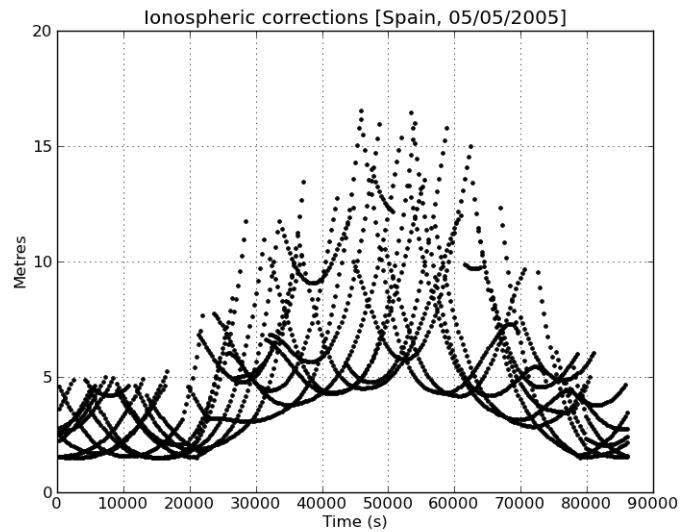


Figure 3.4: Typical Klobuchar values example.

It is also worth commenting that Klobuchar model has some limitations in the southern hemisphere, which are studied in section 5.1.

3.2.2 BeiDou model

This model is equivalent to Klobuchar, but it is designed for BeiDou system. The only differences with Klobuchar are that the layer is at 375 km (25 km higher) and that it uses its own model parameters, generated from monitoring stations in China, which are updated every two hours. These are also broadcasted through the navigation message with the labels “BDSA” and “BDSB” since RINEX version 3.02.

It is stated that BeiDou model outperforms the Klobuchar model for northern hemisphere users in the Asia-Pacific region but exhibits a degraded performance outside this area (source [[Montenbruck and Steigenberger, 2013](#)]).

3.2.3 NeQuick model

NeQuick model has been designed for Galileo system, and it is based on the original profiler developed by [Di Giovanni and Radicella, 1990]. It is a three dimensional and time dependent ionospheric electron density model, which provides the electron density in the ionosphere as a function of position and time. It has five predefined regions (as shown in Fig. 3.5) and has four input parameters that will be broadcasted in header of the navigation message, in the line labeled as "GAL". It also uses 13 data files with numerical values that the receiver must have in order to be able to use the model. These data files are expected to be updated every 5 years approximately, due to its natural variation.

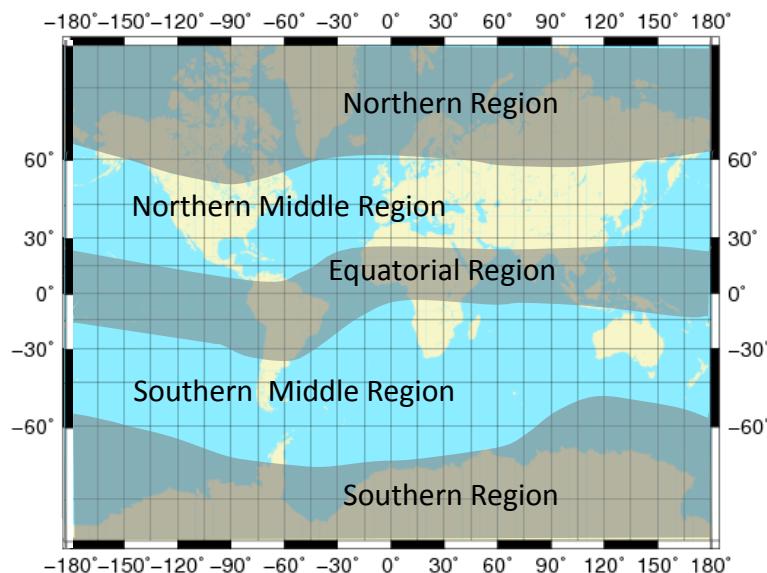


Figure 3.5: The five regions defined in NeQuick model (source [Arbesser-Rastburg, B., 2006])

The algorithm for the Galileo single-frequency receiver is based on the following steps¹ (from [Arbesser-Rastburg, B., 2006]):

1. Model input values are computed using the navigation message parameters.
2. Electron density is calculated for a point along the satellite to receiver path.

¹The algorithm is too long to be explained here, as it is as big as 1/3 of gLAB

3. Steps 1 and 2 are repeated for many discrete points along the satellite receiver path. The number and spacing of the points will depend on the height and they will be a trade-off between integration error and computational time and power.
4. All electron density values along the ray are integrated in order to obtain Slant TEC (or STEC).
5. STEC, in TECUs, is converted to metres of slant delay for correcting pseudo-ranges, by

$$I_f = \frac{40.3 \cdot 10^{16}}{f^2} \text{TEC} \quad (\text{where } f \text{ is in Hz}) \quad (3.17)$$

Note that, as with the Klobuchar model, the ionospheric corrections computed by the NeQuick model can be used for any GNSS signal (GPS, GLONASS, Galileo, etc.) simply by setting the corresponding frequency in (3.17).

3.2.4 GIM (IONEX) model

Global Ionospheric Maps (GIM) is a world-wide map with ionosphere values made by International GNSS Service (IGS). These values are estimated through direct measurements, and are provided as a regular grid. Each grid may have different length between points, and may have or more layers. Each grid is given at a certain epoch in Universal Time (UT), and it may not be at a fixed time period, even though, they are very often published at a rate of 2 hours, with a grid of 2.5 degrees in latitude and 5 degrees in longitude, with a single layer at 450 km, but these values are configurable.

The standard file format used for broadcasting ionospheric maps is known as IONosphere map EXchange format (IONEX)[[Stefan Schaer, 1998](#)] (this is why GIM is often called IONEX). In the header of this file it is given the size of the grid as well as the periodicity. This files are published on a public server the following day, being not provided in real time. Nevertheless, there are predicted GIMs (from two days ahead) in order to allow real time navigation, although the accuracy is worse. These files can be found, for instance, at the public ftp NASA server <ftp://cddis.gsfc.nasa.gov/gps/products/ionex/>.

The algorithm to compute the ionospheric correction is the following:

1. Find the grid maps which are before and ahead of our current time
2. Select the grid map time before our current time
3. Select the first layer in the current grid map
4. Compute the Ionospheric Pierce Point (IPP) in the current grid map and layer
5. Find the four points of the grid which are surrounding the IPP
6. Calculate the Weights (W_i) for each grid point, which is the product of the unitary distance from the IPP to a grid point in each axis (see Fig. 3.6)

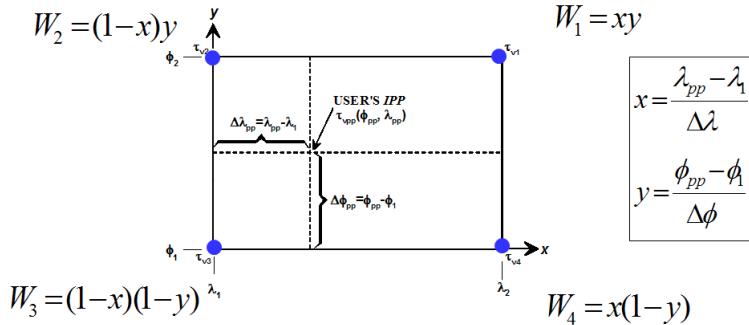


Figure 3.6: Four point interpolation algorithm definition

7. Compute the vertical ionospheric delay, which is the sum of the value of each grid point multiplied by its weight (see equation 3.18)

$$\tau_{vpp}(\phi_{pp}, \lambda_{pp}) = \sum_{n=1}^4 W_i \cdot \tau_{vi} \quad (3.18)$$

8. Compute the mapping function² for the current IPP

$$F = \frac{R_E + h}{\sqrt{(R_E + h)^2 - (R_E \cdot \cos E)^2}} \quad (3.19)$$

Being R_E the Earth radius (provided in the IONEX header), h the layer altitude and E the elevation angle.

²The equation provided corresponds to the “COSZ” function, which is the default if none is specified in the IONEX header. There are others, like the “QFAC” function, but are rarely used. gLAB supports “COSZ” and “QFAC” functions.

9. Calculate the slant delay at the current layer by multiplying the vertical delay with the mapping function
10. Repeat steps 4 to 9 with all the available layers, adding up the slant delay of the previous layer to the current one
11. Calculate the weight of the current grid map time, which is the unitary distance between the current time and the grid map time (like in step 6 but in only 1 dimension)
12. Multiply the slant delay of the current grid map time by its weight
13. Repeat steps 3 to 12 with the grid map ahead of our current time
14. Sum the slant delay from both grid map time. The result is in TECUs
15. STEC, in TECUs, is converted to metres of slant delay for correcting pseudo-ranges, by

$$I_f = \frac{40.3 \cdot 10^{16}}{f^2} \text{TEC} \quad (\text{where } f \text{ is in Hz}) \quad (3.20)$$

Like the previous ionospheric models, it can be used with any GNSS signal by simply setting the corresponding frequency in (3.20).

3.2.5 F-PPP model

The Fast Precise Point Positioning (F-PPP) ionosphere is a real-time world-wide ionospheric model for precise navigation developed by the gAGE/UPC research group. It is still in development, but current tests are giving results about one order of magnitude better than Klobuchar or NeQuick and several times better than the well-known post-processed IGS-GIMs [Rovira-Garcia et al., 2014].

The F-PPP is generated every 5 minutes, and it will be broadcasted through public servers as IONEX files, allowing anyone using the GIM model will be able to use F-PPP. At the current date, it is using a non-standard file format, which gLAB is able to read (see appendix C for details).

Chapter 4

User interface changes and testing

During the update of gLAB, it was intended that all new features available had to be integrated as additional functions, so the main program structure was unchanged. This guarantees full compatibility with older versions, as well as avoiding the user the need to re-familiarize with a new Graphical User Interface (GUI).

Furthermore, not only additional ionosphere models have been implemented, but other secondary improvements have been made. These are the capability of reading version 3.02 RINEX files and the inclusion of ESA's troposphere model. The former is due to that RINEX version 2 files supports GPS, Galileo GLONASS and geostationary constellations, while version 3.02 can handle all of them (including new Japanese QZSS and Chinese BeiDou), so it is highly recommended to use this version (compulsory for multiple constellations files). The latter was done to complete Galileo models, even though gLAB is still unable to process Galileo data, it prepares it for a possible future update on this matter.

4.1 User interface

Here are shown the Graphical User Interface (GUI) tabs which have been changed for the new gLAB version:

4.1.1 Input tab

The Input tab needed two more buttons for IONEX and F-PPP, due to that both of them use external files for their model calculations.

For IONEX data, a IONEX button is available, and it is possible to use DCB correction from IONEX files, as shown in Fig. 4.1.



Figure 4.1: IONEX in gLAB input tab.

The same changes have been made for F-PPP data, as shown in Fig. 4.2.

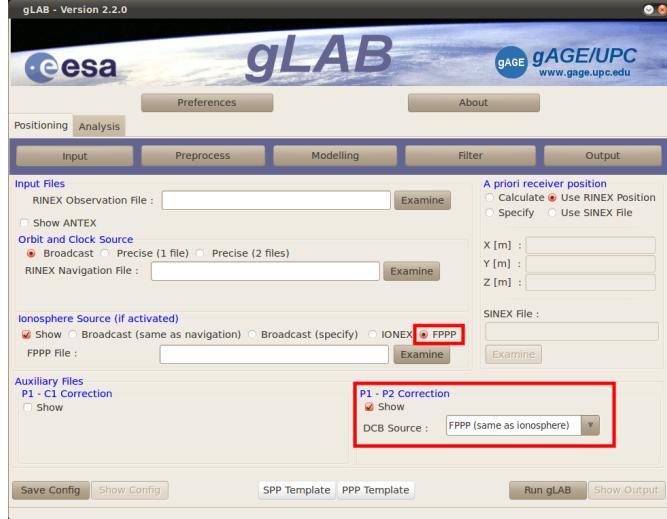


Figure 4.2: F-PPP in gLAB input tab.

It is worth highlighting that in both cases, the DCB's are selected by default from the same ionospheric source when the IONEX or F-PPP button are press, but it is possible to select DCB data from another IONEX or F-PPP file, as shown in Fig. 4.3.



Figure 4.3: DCB sources in gLAB input tab.

4.1.2 Modelling tab

The Modelling tab has not had any significant changes but adding more options in the following corrections options:

The first add-on is in the ionospheric correction options, in which NeQuick, Klobuchar(BeiDou), IONEX and F-PPP models have been added, as shown in Fig. 4.4 (see section 3.2 for ionospheric model explanation).



Figure 4.4: Ionospheric corrections in gLAB modelling tab.

The second add-on is the ESA's tropospheric model (in the tropospheric correction list), as shown in Fig. 4.5. It is worth pointing out that if ESA's tropospheric model is chosen, Niell mapping is selected by default, because this is the only mapping accepted by this model.



Figure 4.5: Tropospheric corrections in gLAB modelling tab.

The third and last change is in the P1-P2 correction options, in which "IONEX file" and "F-PPP file" have been added, as shown in Fig. 4.6.



Figure 4.6: P1-P2 corrections in gLAB modelling tab.

4.1.3 Filter tab

Last but not least, an option in the filter tab has been added to allow the possibility to use the Root Mean Square Error (RMSE) obtained from IONEX or F-PPP files in the filter, as shown in Fig. 4.7.



Figure 4.7: Ionospheric RMSE in gLAB filter tab.

4.2 Testing

Once ionospheric models are implemented in gLAB, they need to be checked to assure their correct functionality. For this purpose, and for each model, an external script bash script was created. All of them have the same code except for the ionospheric model used.

The aim of the script is to test the models in all latitudes and longitudes, so for this purpose, from the station list given, it will sort them in order to make a grid of stations by latitude and longitude (by default, width will be 5° in latitude and 10° in longitude). Once done, the script will process one station in each square (close stations do not provide additional relevant data for this purpose). If the selected station has no observation data, it will try to process other stations until any of them has data or there are not any stations left in that range. The system flow chart is shown in Fig. 4.8.

The execution results are given in three files and four figures for each station:

- Files:
 - ◊ The gLAB output file.
 - ◊ The output file for the reference model (the format may vary in each case).
 - ◊ A data file containing the time (DoY), the ionospheric values for each model and the difference between them.
- Figures:
 - ◊ A figure for gLAB ionospheric model values.
 - ◊ A figure for the reference ionospheric model values.
 - ◊ A figure with both model values superimposed.
 - ◊ A figure with the difference (error) between models.

The most important files are both the difference error data file and figure. The latter allows an easy evaluation of the error value, while the former has the absolute values, where is visible following common patterns (like day/night cycles). The exception is for BeiDou model, where there is no third party program for comparing data, nor even broadcast data. For this reason, BeiDou calculations will be done using Klobuchar's broadcast data of GPS satellites. As both models are very similar, BeiDou results must be very similar too.

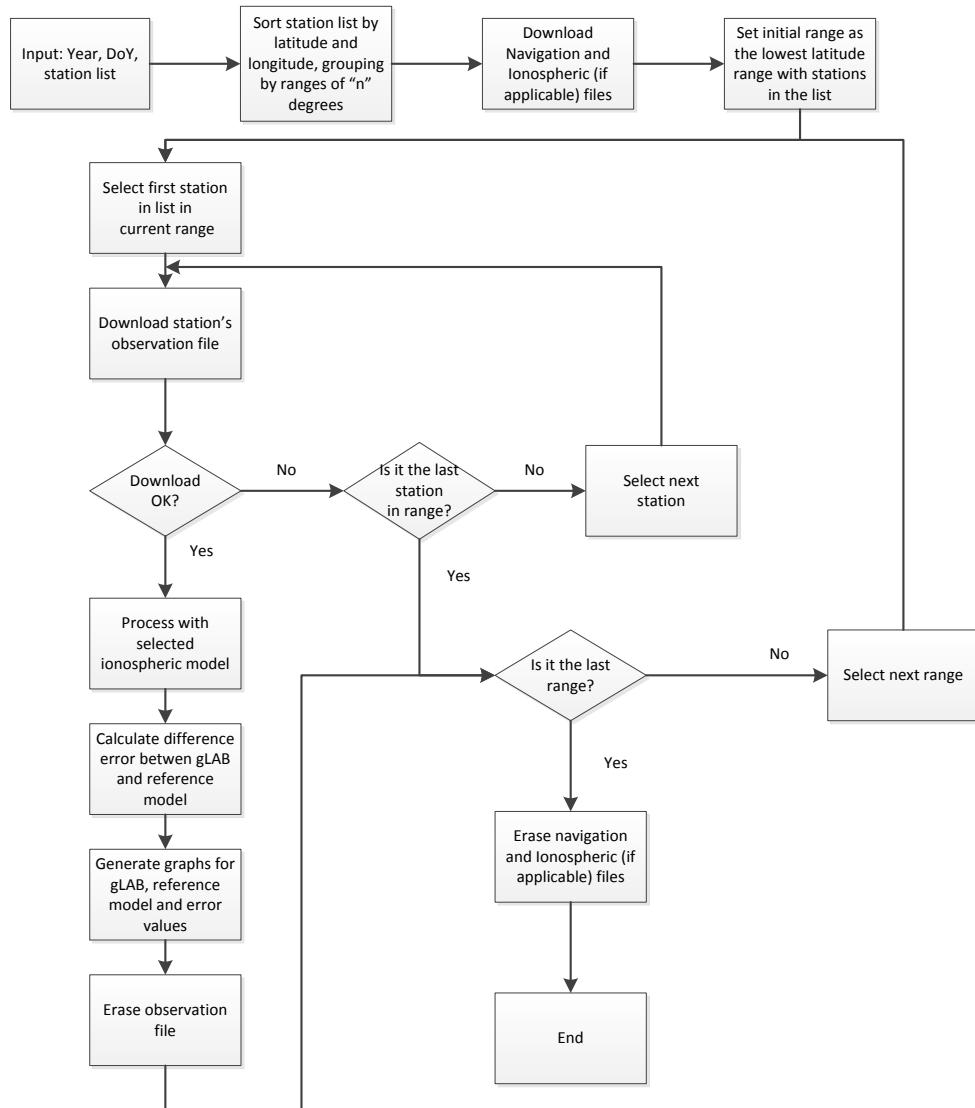


Figure 4.8: Generic test script diagram.

The following images are the difference errors for each model:

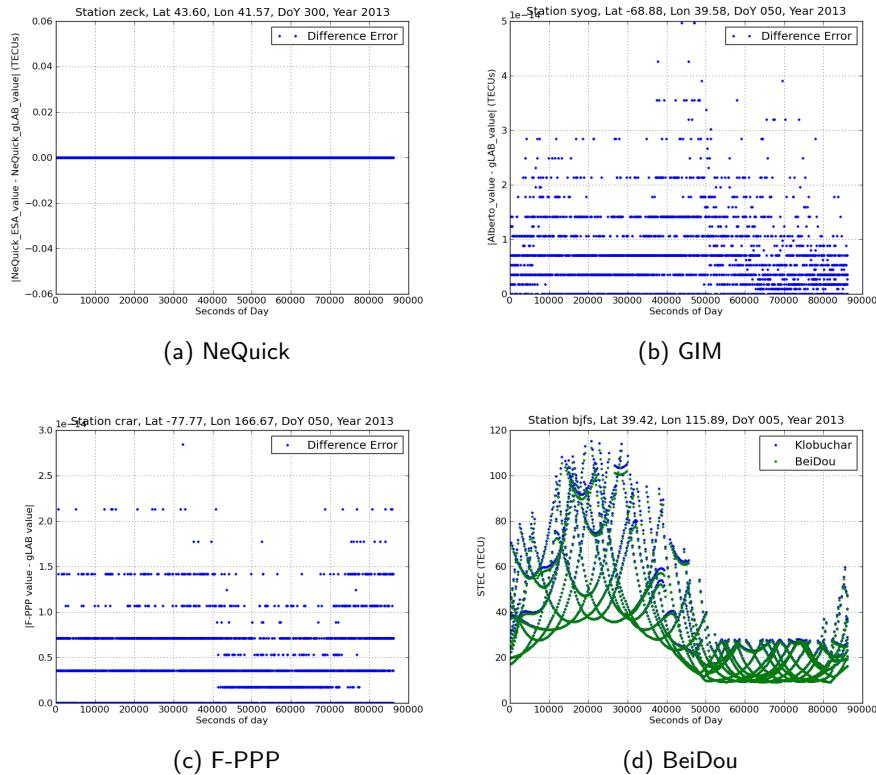


Figure 4.9: Examples of difference between gLAB and reference

The error expected is in the range $[0-10^{-8}]$. This tiny error is due to the limitations when printing the data, where data is rounded in internal calculations. Note that in case of BeiDou the figure is of the model value (with Klobuchar), rather than the difference error, due to there is no reference for BeiDou, but as stated above, the results had to be very close to Klobuchar, which is what the figure shows. The other files are useful for analyzing and tracing errors.

The scripts used to test gLAB are given in electronic format, as well as an instruction guide.

Chapter 5

gLAB applications

The new gLAB version has already been used for two main studies:

5.1 Klobuchar excessive period

As described in section 3.2.1, Klobuchar model assumes there is a constant delay of 5 ns during night time and a cosine function in daytime that is centered at the 14th hour (2 pm) of local time (see Fig 3.2). Therefore, the maximum semiperiod for the cosine function is 20 hours, because the value at the end of the day must be the same at the beginning of the following day (5 ns according to the model).

The semiperiod for the cosine function depends on the user position and the broadcasted ionospheric parameters. It has been detected during gLAB tests that some of the ionospheric parameters broadcasted could lead to a semiperiod greater than 20 hours (see appendix A for a full list of days). In these cases, the ionospheric correction value will have a value gap, due to that the value at 12 p.m according to the previous day is given by the cosine function, but according to the next day it should be 5 ns.

As an example, Hobart station (hob2) will be used, located at Tasmania. This station, due to its geographical location, is one of the most affected by the issue, allowing to see clear gaps in the results. To better illustrate the user, the comparison will be done within the same day in consecutive years, in which the results, in principle, should not have varied much.

In Fig. 5.1 we can see the Klobuchar's ionospheric correction values on the 15th of January in 2011, which is not affected, and in Fig. 5.2 are the same results a year later (the same comparation is made in Fig 5.3 and Fig 5.4 but with satellite number 3):

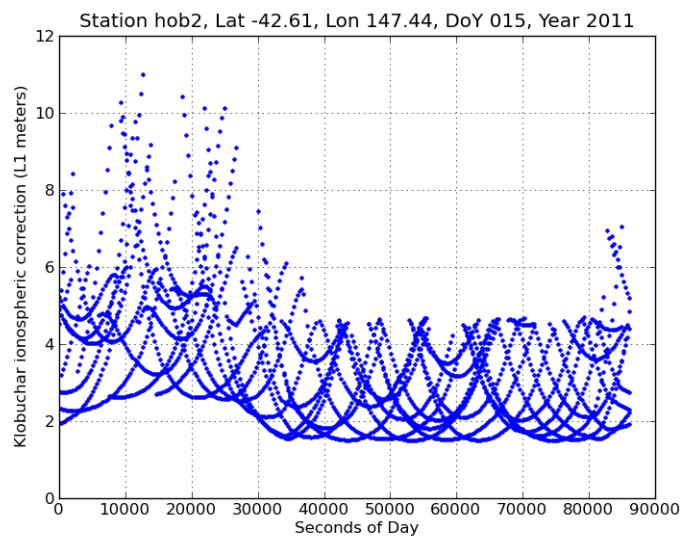


Figure 5.1: Typical Klobuchar ionospheric corrections for all satellites

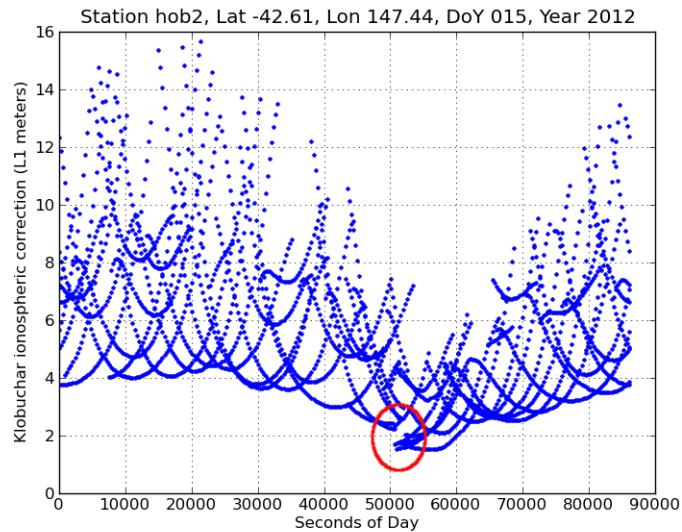


Figure 5.2: Abnormal Klobuchar ionospheric corrections for all satellites

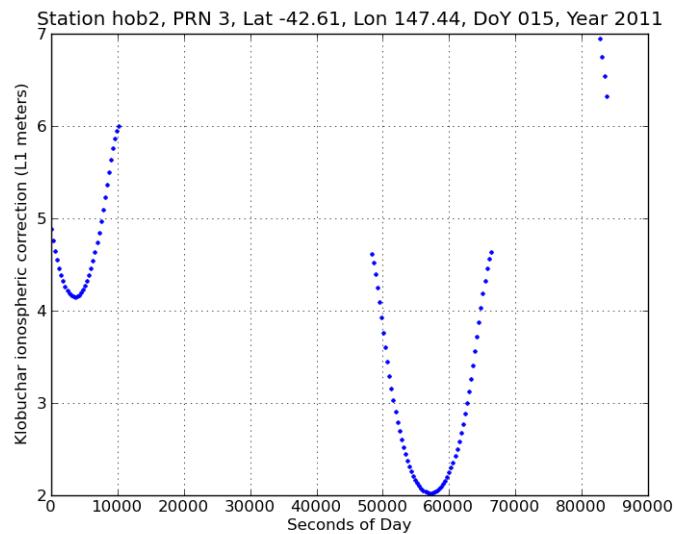


Figure 5.3: Typical Klobuchar ionospheric corrections for a single satellite

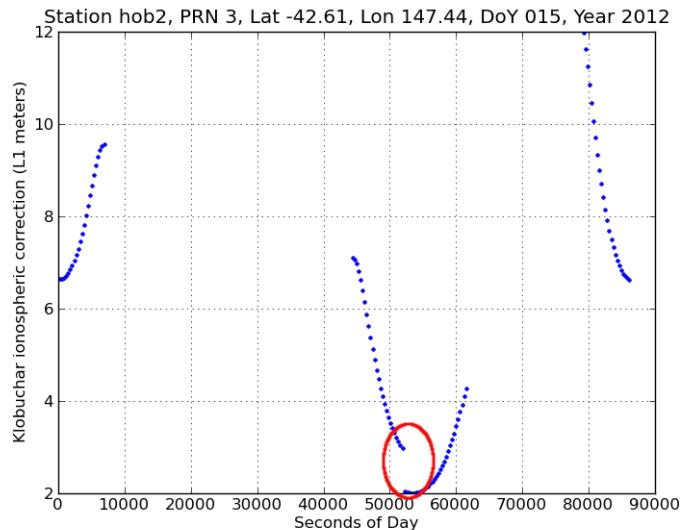


Figure 5.4: Abnormal Klobuchar ionospheric corrections for a single satellite

In Fig. 5.1 it is clearly visible the day/night cycle as the model predicts, where as in Fig. 5.2 there is not such cycle, and there is a clear value gap at over 50.000 seconds (which corresponds to 23:59h in local time in Tasmania), which occurs during the day transition. In this case, the gap is about 1 meter, but it varies according to the given ionospheric parameters and receiver location.

Extending the analysis with all available data (since Jan 1, 1997 till March 20, 2014), there have been many days where this discontinuity has occurred (see table A.2 in appendix A). The size of the gap can be seen in Fig. 5.5 and Fig. 5.6 according to the date when it occurred, and in Fig. 5.7 in function of the geomagnetic latitude.

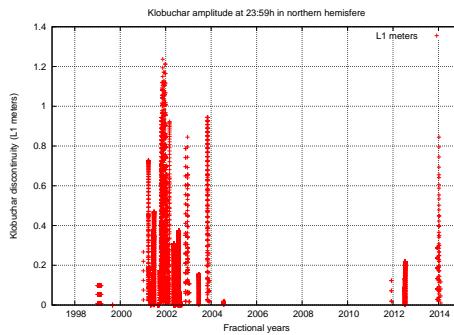


Figure 5.5: Klobuchar gap values during GPS lifetime in northern hemisphere

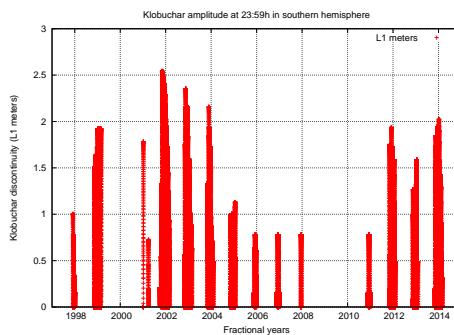


Figure 5.6: Klobuchar gap values during GPS lifetime in southern hemisphere

One of the reasons because the gap value vary so much is due to solar flux. When solar flux is higher, ionosphere perturbations are higher, so the Klobuchar parameters broadcasted are adjusted for higher corrections values, which also lead to greater gaps.

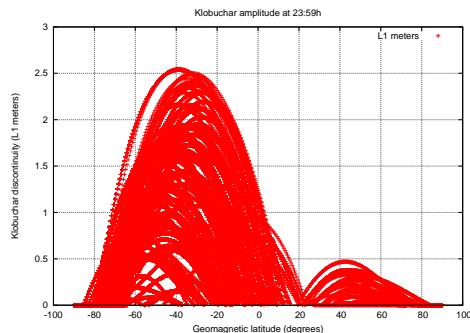


Figure 5.7: Klobuchar gap values during GPS lifetime in function of geomagnetic latitude

As it can be seen in the images above, in the southern hemisphere is where the majority of the discontinuity occurs. This is because on the one side, the formula to calculate the period is more likely to be give higher results for lower latitudes for the given ionospheric parameters, and on the other hand, Klobuchar model is designed to work best at higher latitudes, where the United States of America (USA) are located.

In order to solve this issue, there are three possible solutions:

- **Fix broadcast parameters generation:** The control segment, which generates this parameters, should check that in no case the semiperiod is greater than 20 hours. It has the advantage that only program has to be modified, but it may be necessary to modify the parameter generator.
- **Limit the maximum semiperiod:** The receiver firmware has to be modified in order to put an upperbound of 20 hours in the semiperiod. The advantage is that this modification is very simple to implement, but has the great disadvantage that any existing receiver firmware must be modified, which is nearly impossible (BeiDou model has a semiperiod upperbound, but it is too high to fix this issue).
- **Use cosine values in night time:** The receiver firmware has to be modified in order to use the cosine values during the night until it goes down to the constant value. It is slightly more complex to implement, but it may prove more accurate.

Using the second approach, the data from the 15th of January (Fig. 5.2 and Fig 5.4) will be reprocessed. The results are in Fig. 5.8 and Fig 5.9.

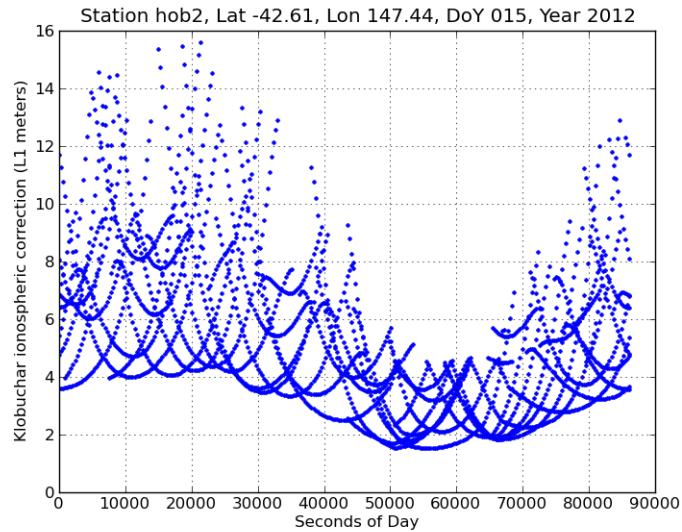


Figure 5.8: Corrected Klobuchar ionospheric corrections for all satellites

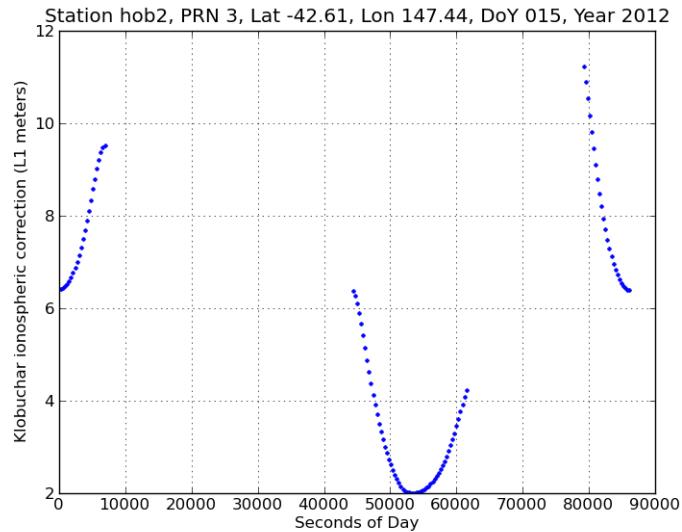


Figure 5.9: Corrected Klobuchar ionospheric correction for a single satellite

In Fig. 5.8, there is no value gap, and there is a clear stable period of 14.400 seconds (4 hours), starting at about 51.000 seconds which corresponds to night time, as the Klobuchar model predicts. The length of four hours is due to that the semiperiod is 20 hours (which determined the length of the day), so the remaining time for one day are the 4 hours stated before.

As for Fig 5.9, there is also no gap, and the values computed are continuous.

Last, but not least, it is necessary to analyze how a semiperiod greater than 20 hours affects user positioning. Thus, it will be compared, for each coordinate, how the error varies between using the nominal Klobuchar model and a corrected version (using the second method for correction, setting the upperbound to 20 hours).

In Fig. 5.10, Fig. 5.11 and Fig. 5.12 below are shown the error obtained in the three coordinates for the nominal and corrected Klobuchar model, among the absolute value of the difference between them:

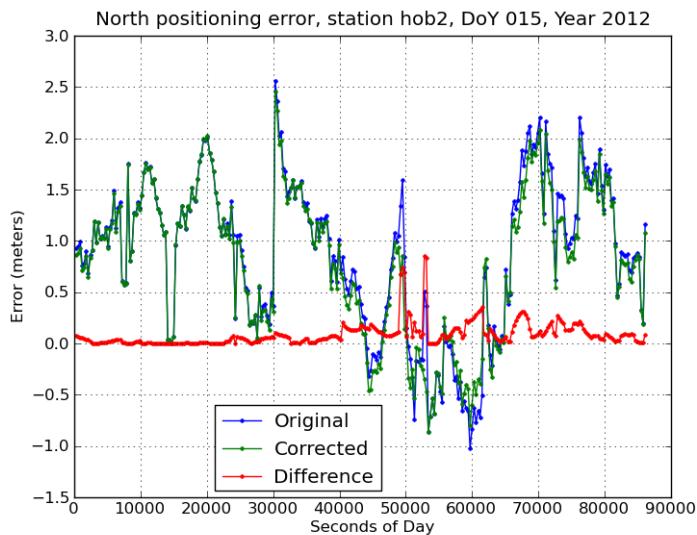


Figure 5.10: Positioning error in north component between original and corrected Klobuchar model

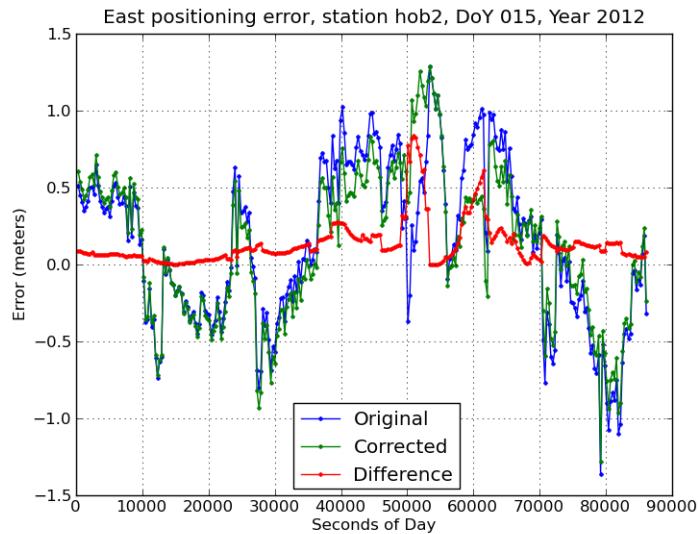


Figure 5.11: Positioning error in east component between original and corrected Klobuchar model

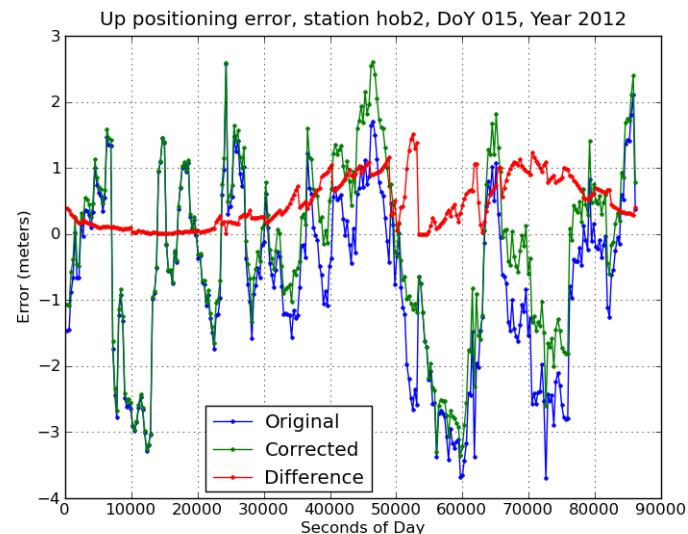


Figure 5.12: Positioning error in up component between original and corrected Klobuchar model

It is clearly visible in all figures that around 50.000 seconds -where the discontinuity occurs- the difference between the models rises to about 1 meter. In the north and up component the corrected model has less error, but the east component has more error than the original, which shows that the corrected model may introduce additional error due to the modified semiperiod.

Although the gap size reaches about a meter, it is not critical, due to that users who use Klobuchar model do simple positioning, where the error is about 5-10 meters. As an example, in Fig. 5.10, Fig. 5.11 and Fig. 5.12, the total error amount oscillates more than a meter, so it is impossible to distinguish if the source of the error is from the Klobuchar model or the few accurate broadcast navigation parameters for computing satellite position or clock drift.

5.2 Ionospheric model comparison

With gLAB capable of processing with multiple ionospheric models, this allows to compare the performance of each model in terms of accuracy (how close is the predicted value from the real one) and error positioning (how ionosphere modelling affects user positioning). In order to acquire precise results it should be necessary to process data from many stations and days, but it is out of the scope for this project. Here it will only be provided a proof of concept, showing the positioning error and RMS for a given day and station.

In the following figures the positioning error is presented for all models (except for BeiDou, due to there were no ionospheric parameters available), with two modes: using only code and using code plus carrier phase. The positioning error is computed using precise orbits, clocks and single frequency. The reason to use precise orbits is to take out the other meter error source, so the results directly show the performance of the ionosphere model.

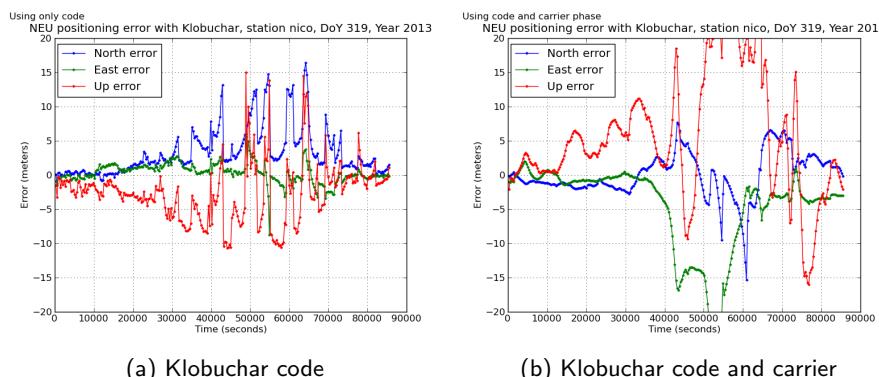


Figure 5.13: Positioning error with Klobuchar model

In both figures Fig 5.13a and Fig 5.13b we can clearly see that after second 40.000 the error increases over 10 meters, even though in Fig 5.13b the error goes even up to 20 meters. As Klobuchar model is not very precise, its error can affect the determination of carrier ambiguities, degrading overall performance. It is also worth to mention that the day where the test was carried (DoY 319, 2013), Klobuchar parameters had the problems stated in section 5.1.

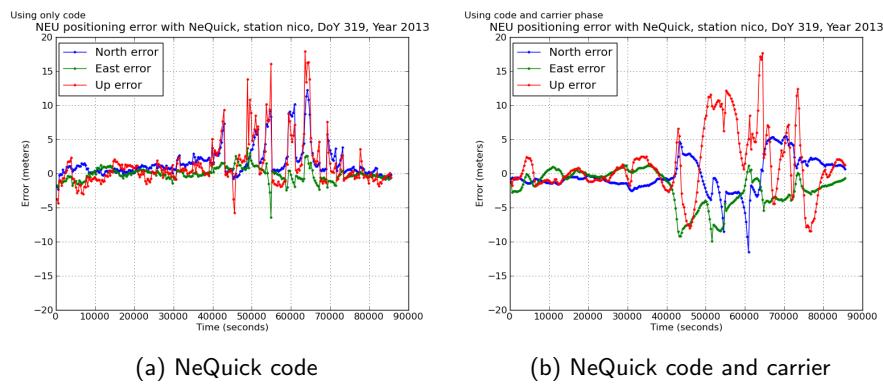


Figure 5.14: Positioning error with NeQuick model

Using NeQuick model, the results seen in Fig 5.14 are very similar in shape with Klobuchar, but NeQuick gets better results. It still has error peaks greater than 10 meters and has greater error when using carrier phase (normally using carrier phase results in fewer error). This means that NeQuick model is also susceptible to other error sources in the same way as Klobuchar.

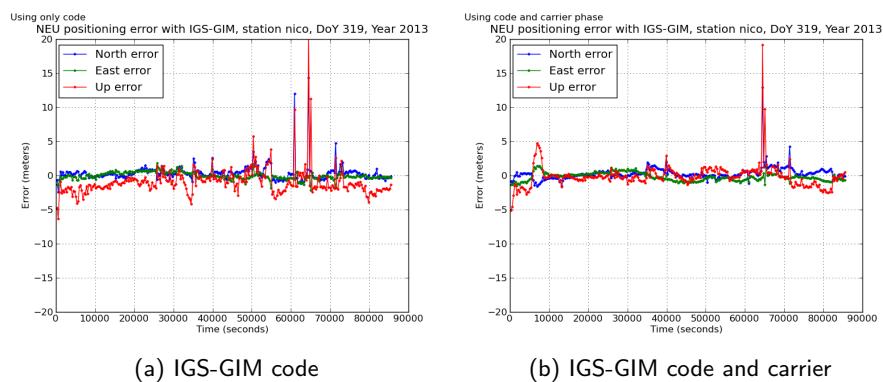


Figure 5.15: Positioning error with IGS-GIM model

IGS-GIM provides much better results as it can be seen in Fig 5.15. The error is around 3 meters, which is the expected for this model. With this model, using code and carrier gets better results, it can be clearly seen in Fig 5.15b, in which the error is smaller and less noisy than in Fig 5.15a. The error peaks can be due to a rare ionospheric disturbance (as at this time the previous models also had error peaks) which the current model is unable to predict.

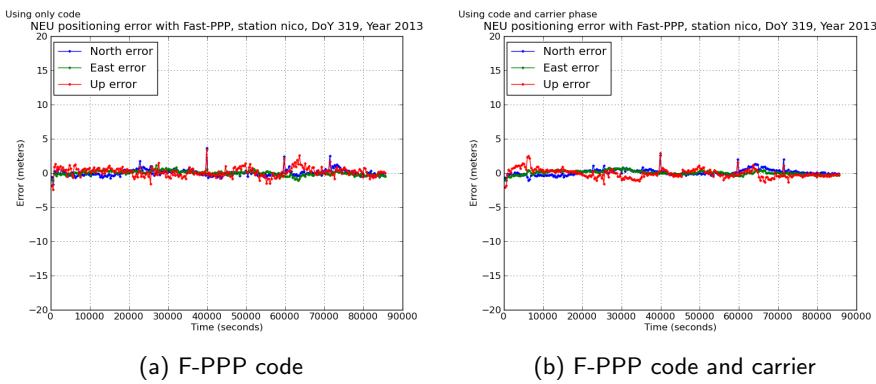


Figure 5.16: Positioning error with F-PPP model

With F-PPP model, it can be seen in Fig 5.16 that the error is at the level of 1 meter, with maximum error peaks of 3 meters. This proves that F-PPP is very precise and capable of following any ionospheric perturbation, due to that it has no error peaks at about 62,000 seconds that appeared in the other models.

The RMS for each figure and the processing time (for the whole process, in gLAB) are in the following tables:

RMS	Klobuchar	NeQuick	IGS-GIM	F-PPP
Code	83.35	63.66	43.02	2.09
Code + carrier phase	344.81	72.53	32.50	1.49

Table 5.1: RMS of 3D positioning error for station hofn, DoY 319, Year 2013

Processing time (s) /day	Klobuchar	NeQuick	IGS-GIM	F-PPP
Code	1.65	4.75	1.71	12.04
Code + carrier phase	1.75	4.96	1.85	12.18

Table 5.2: Processing time of 3D positioning error for station hofn, DoY 319, Year 2013

Moreover, in order to prove that the source of the positioning error showed above are due to the ionosphere model limitations, Klobuchar, NeQuick and IGS-GIM models will be compared with F-PPP model, the only one who was able to reach 1 meter error without large error peaks. The results are in Fig 5.17:

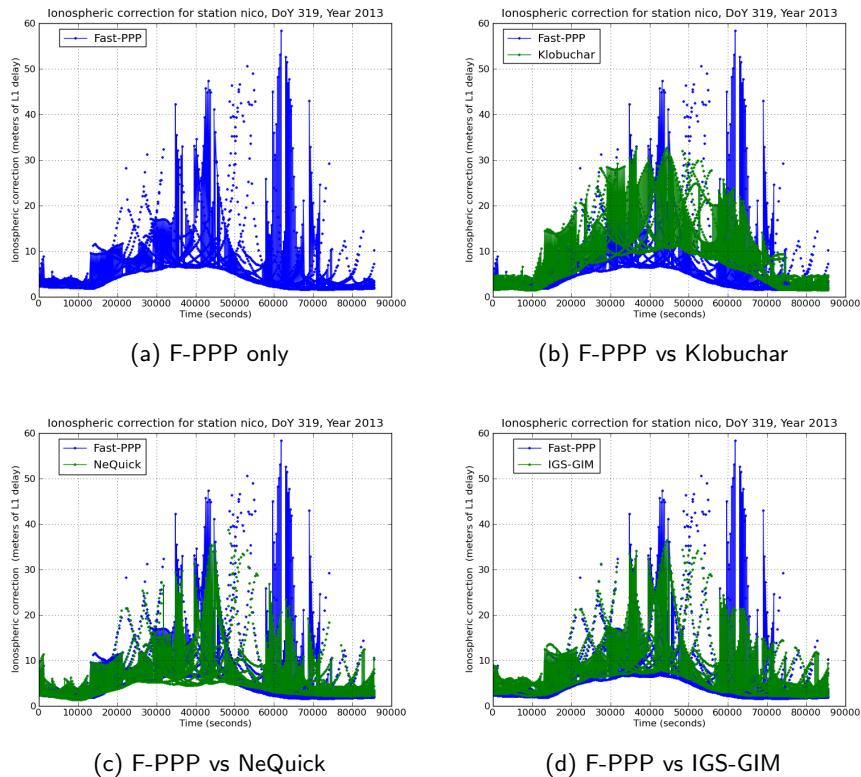


Figure 5.17: F-PPP vs ionospheric models

As it can be clearly seen in Fig 5.17, at 60.000 seconds there are peak values for the ionospheric correction that only F-PPP is able to follow, therefore this is the cause of the positioning error peaks seen at this time in Fig 5.13, Fig 5.14 and in Fig 5.15.

With the results shown above, the best ionospheric model in terms of accuracy is, by far, F-PPP, but at the current state it has the longest processing time due to the file format used, which is very far from using the minimum space as a IONEX does (now is about 100 times bigger than a IONEX file). The disadvantage of F-PPP and IGS-GIM is that they both need an external data file, while Klobuchar and NeQuick only uses the parameters broadcasted in the navigation message. Klobuchar also has the advantage to be the fastest, which makes it more suitable for mobile devices.

5.3 Other studies where gLAB has been used as a Data Processing Tool

In section 5.1 a study of Klobuchar periodicity has been made with gLAB, while in section 5.2 a proof of concept of ionosphere comparisons with gLAB has been made. Additionally, in two proceedings already published the new version of gLAB has been used to compute some of the results presented in them.

In the proceedings [Juan et al., 2014b] and [Sanz et al., 2014] it was computed for several months the ionospheric correction values for Klobuchar, NeQuick, IGS-GIM and F-PPP with gLAB, using these values to make a ionosphere performance comparison during a long period, a direct application shown in section 5.2.

Also, in the proceeding [Vinh et al., 2013], the ionospheric correction values for Klobuchar and IGS-GIM for the region it was studied in the paper (Vietnam) were computed using gLAB.

Last but not least, in the proceeding [Juan et al., 2014a], gLAB was used to compute Klobuchar, NeQuick and IGS-GIM values for a period of about 6 months.

Chapter 6

Conclusion

In this Final Degree Project it has been proven once again that gLAB is a very powerful tool for processing GNSS data. The main goal of the project was to include the ionospheric models described in chapter 3, validate this implementation with the test described in section 4.2 and provide some examples of use for the new version of gLAB.

During the programming phase, it should be emphasised that the source code structure and design of gLAB has proved to have a flexible and fully prepared for updates, although being a source code with about 20.000 lines. Its modular design, with different files for the several parts of the processing (for example a file only for functions for reading files, another only for the filter, etc.), makes it much easier for any programmer to read -and update- the necessary parts.

In the test phase, the results obtained -with the test designed just for this purpose-, proof that the results given by gLAB have the same accuracy up to level of 10^{-9} , hence it can be said the values from gLAB are the same from the third party programs used as a reference. This test was done in a grid of stations in all the world for several days, so the maximum number of possibilities were covered and the results were completely satisfactory.

Finally, in chapter 5, there are two examples of the direct use of gLAB, in section 5.1 there was the Klobuchar period anomaly, which could be thoroughly studied and tested a possible solution modifying the code (where as in proprietary format would not have been possible), and in 5.2 there was a proof of concept of the capability for gLAB for being used to compare ionosphere models, which has proved to be very useful, as it is the main use in the GNSS Meetings and Workshop Proceedings stated in 5.3.

Appendix A

List of days with Klobuchar discontinuity

In this chapter a list of days where Klobuchar model can have a semiperiod greater than 20 hours is presented. Days are grouped in intervals with this issue.

The period analyzed is from 1st of January of 1997 (DoY 1 of Year 1997) to the 20th of March of 2014 (DoY 72 of Year 2014)

Start date (DoY-Year)	End date (DoY-Year)	Number of days
310-1998	338-1998	29
340-1998	341-1998	2
343-1998	352-1998	10
354-1998	364-1998	11
1-1999	1-1999	1
4-1999	9-1999	6
11-1999	48-1999	38
50-1999	58-1999	9
243-1999	243-1999	1
8-2001	8-2001	1
87-2001	95-2001	9
123-2001	123-2001	1
166-2001	180-2001	15
239-2001	258-2001	20
269-2001	276-2001	8
290-2001	59-2002	135
61-2002	61-2002	1
130-2002	134-2002	5
140-2002	157-2002	18
177-2002	178-2002	2
199-2002	217-2002	19
228-2002	239-2002	12
245-2002	246-2002	2
255-2002	257-2002	3
299-2002	48-2003	115
57-2003	57-2003	1

166-2003	168-2003	3
303-2003	303-2003	1
305-2003	311-2003	7
328-2003	346-2003	19
349-2003	24-2004	41
36-2004	37-2004	2
202-2004	206-2004	5
308-2004	315-2004	8
319-2004	343-2004	25
348-2004	9-2005	28
16-2005	23-2005	8
319-2005	319-2005	1
321-2005	330-2005	10
336-2005	351-2005	16
356-2005	1-2006	11
320-2006	325-2006	6
340-2006	351-2006	12
3-2007	3-2007	1
7-2007	8-2007	2
346-2007	352-2007	7
322-2010	323-2010	2
322-2010	323-2010	2
342-2010	345-2010	4
347-2010	350-2010	4
3-2011	7-2011	5
309-2011	30-2012	87
32-2012	36-2012	5
186-2012	186-2012	1
189-2012	189-2012	1
192-2012	194-2012	3
196-2012	197-2012	2
317-2012	20-2013	70
36-2013	36-2013	1
303-2013	304-2013	2
311-2013	61-2014	116

Table A.1: List of days with Klobuchar discontinuity

In the following table there is a summary with the total number of days with discontinuity, and the percentage of these in the days processed.

	Number of days with discontinuity	Number of GPS days processed	Percentage
In northern hemisphere	347	6289	5,52%
In southern hemisphere	888	6289	14,12%
In any latitude	1001	6289	15,92%

Table A.2: Percentage of days with Klobuchar discontinuity

Appendix B

Multi-platform arrangements

One of European Space Agency (ESA)'s requirements is that gLAB tool should be multi-platform. gLAB was developed in Linux due to the powerful command line shell available, which is perfect for scripting, fast data handling and mass processing. In order to show these capabilities and how gLAB was designed to be used, all the tutorials and exercises were prepared to be done under a Linux environment. Therefore, in non Linux platforms it is necessary to add additional software so as the user experience is the same as in Linux.

B.1 Windows

There is an application for Windows called Cygwin (<https://cygwin.com/>), developed by RedHat and published under GPL license, which creates a Linux shell in Windows. Their slogan is “Get that Linux feeling... in Windows” and they got it. It has most of the programs that a Linux user uses, and it can run Linux shell scripts straight out of the box. For our purposes, this allows to use all the training and exercises material for Linux to be run under Windows without modifications.

For an easy start for any user, an automatic installer (and uninstaller) for Cygwin has been created and embedded in gLAB's installer. The steps to do it are the following:

- **Retrieve Cygwin packages:** Cygwin has a lot of packages, but only the necessary ones will be installed in order to save disk and time.

- **Cygwin installer batch script:** Create a batch script (our script is “Install Cygwin.bat”) for installing Cygwin which will receive as a parameter the gLAB installation path (from the installer). The script must be able to work with any language and Windows version. The steps in the script are:
 - ◊ Call Cygwin installer in unattended mode, giving as parameters the installation path (in our case fixed to C:\Cygwin), and the list of packages to install.
 - ◊ Retrieve the desktop and start menu path from the registry and create shortcuts
 - ◊ Copy Cygwin uninstall script to a system path. It cannot be copied to gLAB’s installation directory because the user must be able to uninstall gLAB without uninstalling Cygwin or erasing our Cygwin uninstaller script.
 - ◊ Copy bash user profiles to user’s home directory (this is optional, but it sets our aliases and colour configuration we like in Cygwin)
 - ◊ Copy gLAB’s and graph execution scripts to Cygwin folder. These scripts are due to that gLAB and graph programs used are the Windows version, but in the exercises they will be called with linux paths, so these scripts automatically convert to Windows paths.
 - ◊ Modify the scripts from the previous step in order to include in them the path where gLAB is installed. In this case the installer called a bash script (see items below).
- **Cygwin uninstaller batch script:** Create a batch script for uninstalling (the file is “Uninstall Cygwin.bat”). There is no official uninstaller for Cygwin (only the instructions on the web page), so creating a script was necessary. It must be also able to work with any language and Windows version. The steps in the script are:
 - ◊ Check for administrative privileges. If it is called from gLAB’s uninstaller it will inherit administrative privileges, but if it is called from the Start Menu it will not have them. It should prompt the user for privileges if it has not got administrative privileges.
 - ◊ Prompt the user to confirm if he wants to delete Cygwin. This allows the user to uninstall gLAB but not Cygwin, and also as a confirmation in case the user did not want to call the uninstaller.
 - ◊ Retrieve the desktop and start menu path from the registry and delete shortcuts and user profiles files copied in the installation script.
 - ◊ Delete Cygwin folder and Cygwin data in the Windows registry.
 - ◊ Delete the uninstaller script file itself, so there is no files remaining in the computer.

- **Cygwin configuration files:** As mentioned in the Cygwin installer script, if the prompt shell is to be customized, the files `.bashrc`, `.bash_profile`, `.minttyrc` and `.vimrc` have to be created and copied to the home folder.
- **gLAB's and graph execution scripts:** As mentioned in the Cygwin installer script, these execution scripts just read the input parameters and check for pathes. If a Linux path is found, it translates to Windows paths. Afterwards they call the Windows gLAB or graph executables with the given parameters (our file is “`.run_graph.sh`”).
- **Insert path in execution scripts:** The execution script needs to know where gLAB is installed, which a priori is not known. For this reason, during installation a bash script was created in order to insert the path in them (the file is “`Configure_cygwin.sh`”). Additionally, this execution scripts were linked to the “`/bin`” folder, so they can be accessed from any directory.
- **Modify gLAB's installer script:** gLAB's installer is created with an Inno Setup script. It has to be modified so it includes the Cygwin installation files, copy them to a temporal folder (afterwards these files are no longer needed), show the option to install Cygwin (using the installation script) after gLAB has installed and call the Cygwin uninstall script when uninstalling.

All the previous steps require some scripting knowledge, in both Windows and Linux and it is out of the scope of this project to explain how to do them. Nevertheless, all scripts have plenty of comments and are included in the electronic content, so any user can make his own installer.

B.2 Macintosh

After several tests, it was not possible to make the GUI and the graph module to work under OSX due to python problems. Therefore, the solution for Macintosh is to create a virtual machine with VirtualBox, containing a light version of Ubuntu, with gLAB already installed. This allows any user in any platform to download the virtual machine (shared in a standard “`.ova`” format), open it and start working in a few minutes. The virtual machine files and the installation instruction are available at gAGE's website http://www.gage.es/gnss_book.

Appendix C

F-PPP v0.50

This is a preliminary unofficial version file for the F-PPP model. It has a header with the same pattern as the RINEX. Once the header ends, a data map starts with a list of hardware delays for each satellite (the order of the data is given in the header, and each TEC may have their own hardware delays, so the last one found should be used). Then a ionospheric map is given for a determinate region and for a given epoch (every 5 minutes), ordered by longitude and latitude. The five fields in each line are: longitude, latitude, the partial Total Electron Content (TEC), the sigma (mean error) and the map layer (the ionosphere is given in two layers).

(Continues in the next page)

0.5	IONOSPHERE MAPS					FPPP	FPPP VERSION / TYPE
2013 11 15 0 0 0							EPOCH OF FIRST MAP
300							INTERVAL
6371.0							BASE RADIUS
268.0 1600.0							HGT1 / HGT2
2.5 2.5							DLT / DLAT
-90 90							MIN/MAX LAT
TECUS							MAP UNITS
PRN FRAC_1 FRAC_W IFB(P1-P2)							HARDWARE DELAYS
cycles							FRAC UNITS
L1-L2 (LI) meters							IFB UNITS
1							END OF HEADER
G 1 0.2529 0.1434 -4.320							START OF TEC MAP
G 2 -0.2733 0.0176 1.201							
G 3 0.5647 0.5431 -1.847							
G 4 -0.6354 0.2289 -1.545							
G 5 0.7510 -0.4209 -0.836							
G 6 -0.1205 -0.2831 -1.750							
G 7 -0.0502 0.0081 -0.648							
G 8 -1.3232 0.4370 -1.903							
G 9 -0.1151 0.2681 -1.631							
G10 -0.2813 -0.2179 -2.269							
G11 -0.2955 0.3953 -0.459							
G12 0.5937 0.3640 -0.424							
G13 1.2399 -0.1631 -0.615							
G14 0.0338 0.0258 -0.962							
G15 0.1745 0.1818 -0.808							
G16 0.2792 0.1751 -0.768							
G17 0.0090 0.2541 -0.744							
G18 -0.0292 0.3420 -0.583							
G19 0.0732 0.1530 0.153							
G20 0.0634 -0.2927 -1.159							
G21 0.8965 -0.4240 -0.561							
G22 1.1177 -0.0109 0.681							
G23 0.5559 -0.0780 1.135							
G24 0.2802 -0.1223 -3.347							
G25 0.4423 -0.1454 -3.799							
G26 -0.3572 0.0590 -1.616							
G27 -0.3993 -0.3083 -3.100							
G28 -0.1628 0.4581 -0.696							
G29 -0.2691 0.0889 -0.968							
G31 0.0087 0.2139 -0.162							
G32 -0.3182 0.1141 -2.102							
2013 11 15 0 0 0							EPOCH OF CURRENT MAP
0.0 -90.0 13.7238 1.43e+00 2							
2.5 -90.0 13.6190 1.33e+00 2							

5.0	-90.0	13.5047	1.24e+00	2
7.5	-90.0	13.4000	1.14e+00	2
10.0	-90.0	13.2857	1.05e+00	2
12.5	-90.0	13.1714	9.52e-01	2
15.0	-90.0	13.0666	9.04e-01	2
17.5	-90.0	12.9524	8.66e-01	2
20.0	-90.0	12.8476	8.37e-01	2
22.5	-90.0	12.7333	8.37e-01	2
25.0	-90.0	12.6190	8.56e-01	2

...

355.0	90.0	4.2798	6.21e+01	1
355.0	90.0	4.5782	5.99e-01	2
357.5	90.0	4.0436	6.49e+01	1
357.5	90.0	4.5592	6.38e-01	2

2

START OF TEC MAP

G 1	0.2426	0.1439	-4.320
G 2	-0.2832	0.0175	1.202
G 3	0.5537	0.5429	-1.847
G 4	-0.6443	0.2294	-1.545
G 5	0.7331	-0.4201	-0.836
G 6	-0.1312	-0.2831	-1.749
G 7	-0.0576	0.0081	-0.648
G 8	-1.3331	0.4371	-1.903
G 9	-0.1244	0.2683	-1.631
G10	-0.2864	-0.2180	-2.268
G11	-0.3059	0.3952	-0.459
G12	0.5937	0.3640	-0.424
G13	1.2203	-0.1641	-0.615
G14	0.0244	0.0251	-0.963
G15	0.1708	0.1817	-0.808
G16	0.2723	0.1750	-0.768
G17	0.0090	0.2541	-0.745
G18	-0.0372	0.3417	-0.584
G19	0.0620	0.1530	0.153
G20	0.0565	-0.2925	-1.159
G21	0.8867	-0.4242	-0.563
G22	1.1336	-0.0178	0.681
G23	0.5459	-0.0781	1.135
G24	0.2712	-0.1229	-3.346
G25	0.4386	-0.1471	-3.801
G26	-0.3603	0.0590	-1.617
G27	-0.4131	-0.3091	-3.100
G28	-0.1725	0.4584	-0.695
G29	-0.2774	0.0886	-0.967
G31	999.0000	999.0000	-0.162
G32	-0.3295	0.1140	-2.102

2013 11 15 0 5 0

EPOCH OF CURRENT MAP

0.0	-90.0	13.7808	1.43e+00	2
2.5	-90.0	13.6760	1.33e+00	2
40.0	7.5	6.5933	1.14e+01	1
40.0	7.5	11.1141	4.38e+00	2
42.5	7.5	1.2745	1.71e+01	1
42.5	7.5	11.0098	3.90e+00	2
45.0	7.5	6.9087	1.43e+01	1
77.5	7.5	-1.5044	1.24e+01	1
77.5	7.5	8.5045	5.99e+00	2
80.0	7.5	1.5689	6.57e+00	1
80.0	7.5	8.8855	8.28e+00	2
82.5	7.5	2.1129	6.19e+00	1
87.5	7.5	9.8379	5.23e+00	2
90.0	7.5	1.7597	5.70e+00	1
90.0	7.5	10.2577	7.89e+00	2

...

Acknowledgements

I would also like to thank my family for their patience and support during this time.

I would like to specially thank my project director Dr. Jaume Sanz Subirana for teaching me GNSS theory from scratch and his continuous help during the realisation and writing of this project.

I would like to thank Adrià Rovira Garcia and Dr. José Miguel Juan Zornoza for their continuous support during all the project.

Last, but not least, thanks to Dr. Alberto García Rigo for his help when testing IONEX model in gLAB, while listening to my heavy metal music playlists.

Bibliography

- [Arbesser-Rastburg, B., 2006] Arbesser-Rastburg, B., 2006. The Galileo Single Frequency Ionospheric Correction Algorithm. <http://sidc.oma.be/esww3/presentations/session4/arbesser.pdf>.
- [Di Giovanni and Radicella, 1990] Di Giovanni, G. and Radicella, S., 1990. An Analytical Model of the Electron Density Profile in the Ionosphere. *Advances in Space Research* **10**(11), pp. 27–30.
- [Juan et al., 2014a] Juan, J., Sanz, J., González-Casado, G., Rovira-Garcia, A. and Ibáñez, D., 2014a. Accurate reference ionospheric model for testing GNSS ionospheric correction in EGNOS and Galileo.
- [Juan et al., 2014b] Juan, J., Sanz, J., Rovira-Garcia, A., González-Casado, G., Escudero Royo, M., Ibáñez, D., Vinh, L. T. and Khoi, V., 2014b. An Accurate Global Ionospheric Model for GNSS users.
- [Klobuchar, 1987] Klobuchar, J., 1987. Ionospheric Time-Delay Algorithms for Single-Frequency GPS Users. *IEEE Transactions on Aerospace and Electronic Systems* **AES-23**(3), pp. 325–331.
- [Montenbruck and Steigenberger, 2013] Montenbruck, O. and Steigenberger, P., 2013. The BeiDou Navigation Message. International Global Navigation Satellite Systems Society, IGNSS Symposium 2013 pp. 8–9.
- [Rovira-Garcia et al., 2014] Rovira-Garcia, A., Juan, M. and Sanz, J., 2014. A Real-time World-wide Ionospheric Model for Single and Multi-frequency Precise Navigation. In: Proceedings of 27th International Technical Meeting of The Satellite Division of the Institute of Navigation, Tampa, Florida (USA).
- [Sanz et al., 2013] Sanz, J., Juan, J. and Hernández-Pajares, M., 2013. *GNSS Data Processing, Vol. I: Fundamentals and Algorithms*. ESA Communications, ESTEC. TM-23/1., Noordwijk, the Netherlands.

- [Sanz et al., 2014] Sanz, J., Rovira-Garcia, A., Juan, J., González-Casado, G., Ibáñez, D. and Escudero, M., 2014. Fast-PPP on a planetary scale.
- [Stefan Schaer, 1998] Stefan Schaer, W. G., 1998. IONEX: The IONosphere Map EXchange Format Version 1. In: Proc. of the IGS Analysis Center Workshop., Darmstadt, Germany.
- [Vinh et al., 2013] Vinh, L., Quang, P. X., Garcia-Rigo, A., Rovira-Garcia, A. and Ibáñez Segura, D., 2013. Experiments on the Ionospheric Models in GNSS. In: Proceedings of 20th Asia-Pacific Regional Space Agency Forum, Vol. 113number 335, Hanoi, Vietnam.